DS

چاپی نوێ

2018

شێوازی پێکهاتنی زانیاریی

نووسینی :

هێمن مەلا کەریم بەرزنجی

ناسنامەی کتێب

ناوی کتێب : شێوازی پێکهاتنی زانیاری بە بەکارهێنانی جاڤا.

نوسینی : هێمن مەلا کەریم بەرزنجی

پیتچنین و نەخشەسازی : هێمن کەریم ئەحمەد

تیراژ : 2000 دانە

نۆبەتی چاپ : یەکەم، دووەم، سێهەم

ساڵی چاپ : 2015، 2016، 2018

Mobile: 07701515582

E-Mail:

dr.hemn@yahoo.com

Facebook:

facebook.com/ hemn.melakarimbarznji

مافی لە چاپدانەوە، بڵاوکردنەوە، لەبەرگرتنەوە و هەموو جۆرە بەکارهێنانێکی پارێزراوە بۆ نووسەر

شێوازی پێکهاتنی زانیاریی

Data Structure

نووسینی: هێمن مەلا کەریم بەرزنجی

چاپـــــی سێهەم - 2018

پێشکەشە بە:-

- **ڕۆحی مەرحومی باوکم ((حاجی بابا شێخ عەبدولکەریمی بەرزنجی)) .**
- **یەکەم مامۆستای ژیانم، ڕۆحی شادبێت و خوای گەورە لێی خۆشببێت.**
- **دایکم و هەردوو براکەم و هاوسەری ژیانم ((چڕۆ)) و موحەمەدی کوڕم.**
- **هەموو ئەو مرۆڤانەی بێوچان و بەردەوام لە هەوڵی فێربوون و، بەخشینی زانستدان.**
- **هەموو ئەو مامۆستایانەی لە حوجرەو قوتابخانەکانەوە، تا ئەمڕۆ، تەنها وشەیەك چییە لێیانەوە فێر بووم.**
- **ئەو مرۆڤەی دەبێتە هاوسەری ژیانم، خۆشم دەوێ و خۆشی دەوێم؟!!**

پێشەکی چاپی سێهەم :

شێوازی پێکهاتنی زانیاری Data Structure ، یەکێك لە گرنگترین بوارەکانە بۆ تێگەیشت لە کۆمپیوتەر و تەکنەلۆژیای زانیاری و، بایەخێکی زۆر گەورە و گرنگی هەیە لە بنیاتنانی نەزمی کارپێکردنی کۆمپیوتەر ((نەزمی بەگەڕ خستن Operating System)) و بەرنامەسازیی Programming و، درووستکردنی بەرنامە Software Development و، زۆربەی بوارەکانی تردا. هەربۆیە فێربوونی بۆ بەرنامە ساز Programmer و ئەندازیار Engineeer و هەموو پسپۆرەکانی تری بواری تەکنەلۆژیا پێویستە، فێربوونیش بە زمانی دایك ئاسانترە و، کاتێکی کەمتریشی دەوێت، ئەمە جگە لەوەی کە نەوەی نوێ زمانە بیانییەکان بە باشی و ، وەك پێویست نازانن و، ناتوانن بە زمانە زیندو و باوەکانی وەك ئینگلیزیی و عارەبیی و فارسی و فەرەنسی و ئالمانی و .... بخوێننەوە، بۆیە لای خۆمەوە دەستم بە نووسینی شێوازی پێکهاتنی زانیاری Data Structureکرد و، لەگەڵ وتنەوەی هەر بەشێکی بابەتەکە لە کۆلێژ، بەشێکم بە زمانی شیرینی کوردیی لێنووسی و، بەمەش کتیبێکی 500 لاپەرەی ئەکادیمی زانستی هاتە بەرهەم، کە هاوکات شێوازی پێکهاتنی زانیاری بە بەکارهێنانی زمانی بەرنامە سازیی جاڤا بە بەرنامەی نێتبین Java Programming Language using Netbean جێبەجێکراوە.

لەم بەرهەمەدا، هەوڵمداوە سەرجەم وشە و زاراوە ئینگلیزییەکانیش ، لە پاڵ وشە و زاراوە کوردییەکان دا بنووسمەوە و، بەمەش هەم سوودی زیاتر بە خوێنەر دەگات و، هەم خزمەت بە زمانی کوردیی دەکەین و، لە بواری وشەو زاراوەی تەکنەلۆژیی دا بەرەو پێشی دەبەین.

ئەم بەرهەمەش وەك سەرجەم بەرهەمەکانی تر، بە هەوڵ و مانوو بوون و لێبڕان بەرهەم هاتووە و، کاتێکی زۆری پێبەخشراوە، بەڵام لەگەڵ ئەوەشدا بێکەم و کورتی نییە و، هەڵەی تێدایە. چونکە کاری مرۆڤە و ، مرۆڤیش لە هەڵە بەدەر نییە. بۆیە سوپاسی هەر رەخنەیەکی بنیاتنەر دەکەم .

لە زنجیرە بابەتێکدا کە لە چەند بەشێك پێکهاتووە ناوەرۆکی ئەم کتیبەتان دەخەمە بەردەست، بەو هیوایەی سوودێك ببەخشێت بە کتیبخانەی کوردیی و، زمانی کوردیی و، سەرجەم فێرخوازانی کورد.

هێمن مەلا کەریم بەرزنجی ، کوردستان سلێمانی ، مامە یارە

شێوازی پێکهاتنی زانیاریی

Data Structure

نووسینی: هێمن مەلا کەریم بەرزنجی

بەشی یەکەم

بنچینەکانی شێوەی

پێکهاتنی زانیاری

Data Structure

Basics

شێوەی پێکهاتنی زانیاری

Data Structure

لە زاراوەکانی کۆمپیوتەر دا، شێوەی پێکهاتنی زانیاری Data Structure ڕێگەیەکی دیاری کراوە بۆ پاشەکەوت کردن Store و ڕێکخستنی Organize زانیاری لە یادگەی کۆمپیوتەر Computer's Memory کە ئەم زانیارییە دەگونجێت بەکاربهێنرێت بە شێوەیەکی چالاك و تەواو لە کاتێکی تر دا و، جێبەجێکردنی هەموو کردارەکان لەسەر زانیارییەکە. دەتوانرێت زانیاری ڕێکبخرێت لە چەندین ڕێگە و شێوەی جیاواز دا.

پۆلێنەکانی شێوەی پێکهاتنی زانیاری

Categories of Data Structure

شێوەی پێکهاتنی زانیاری Data Structure دابەشدەکرێت بۆ دوو شێوەی سەرەکی، ئەم پۆلێن بەندییەش ئەمانەن:

یەکەم// شێوەی پێکهاتنی زانیاری هێڵی

Linear Data Structure

بە شێوەی پێکهاتنی زانیاری Data Structure دەوترێت هێڵی Linear، ئەگەر دانە پێکهێنەرەکان دیاری کرابن بۆ شێوە و ڕیز بەندییەکی دیاری کراوو. بۆئەمەش دوو شێوازی هونەری بنچینەیمان هەیە بۆ پیشاندانی شێوەی پێکهاتنی هێڵی Linear Structure لە یادگەی کۆمپیوتەر دا.

- ڕێگەی یەکەم داببنکردنی پەیوەندی هێڵییە لە نێوان، هەموو دانە پێکهێنەرە پیشاندراوەکان دا، بە سوود وەرگرتن لە شوێنی یادگە و پاراستنی هێڵی Linear Memory، ئەم شێوازی پێکهاتنی هێڵییە وەکو ڕیزکراوەکانە Array.

- ڕێگەی دووەم، دابینکردنی پەیوەندی هێڵییە لە نێوان هەموو دانە پێکهێنەرە پیشان دراوەکان دا، بە بەکارهێنانی بیرۆکەی هێماپێدەرەکان ((ئاماژەپێدەرەکان Pointer)) و بەستەر Link. ئەم شێوازی پێکهاتنی هێڵییە وەکو لیستەی پێکەوە بەستراوو Linked List وایە.

نموونە باوەکانی شێوەی پێکهاتنی هێڵی ئەمانەن :

- ریزکراوەکان Arrays.
- لیستەی پێکەوە بەستراوو Linked List.
- شارای نۆبە بەستن Queues.
- شارای ڕەفەیی Stack.

دووەم// شێوەی پێکهاتنی زانیاری نا هێڵی

Non-linear Data Structure

ئەم شێوەی پێکهاتنە بە زۆری بەکاردێت بۆ پیشاندانی زانیاری کە پەیوەندی هەرەمی hierarchical relationship تێدایە لە نێوان دانە پێکهێنەرە جیاوازەکان دا.

نموونەی شێوەی پێکهاتنی زانیاری ناهێڵی وەکو ئەمانەی لای خوارەوە:

1- گرافەکان Graphs.
2- خێزانی درەختەکان Family of Trees.
3- خشتەی ناوەڕۆکەکان Table of Contents.

تایبەتمەندییەکانی بنکەی زانیاری

Properties of Data Structure

تایبەتمەندیی و، ڕووخسارەکانی شێوەی پێکهاتنی زانیاری زۆرن، بەڵام لێرەدا تەنها گرنگترین تایبەتمەندییەکان دەخەینە ڕوو:

- راستێتی Correctness: جێبەجێکردنی شێوەی پێکهاتنی زانیاری پێویستە جێبەجێکردنی ڕووکارێك بێت بە شێوەیەکی ڕاست.
- ئالۆزێتی کات Time Complexity: پێویستە کاتی جێبەجێکردن یان کاتی جێبەجێکردنی کردارەکانی شێوەی پێکهاتنی زانیاری بچوك و کەمترین بێت، هێندەی بگونجێت و شیاوو بێت.
- ئالۆزێتی بۆشایی Space Complexity: یادگەی بەکارهێنراوو بۆ کردارەکانی شێوەی پێکهاتنی زانیاری پێویستە کەمترین و بچوکترین بێت، هێندەی بگونجێت و، شیاوو بێت.

دۆخەکانی کاتی جێبەجێکردن

Execution Time Cases

بە شێوەیەکی باو، سێ دۆخ هەیە بۆ بەراورد کردنی کاتی جێبەجێکردنی جۆرە جیاوازەکانی شێوەی پێکهاتنی زانیاری ، کە ئەمانەن:

- خراپترین دۆخ Worst Case: کاتێك جیبەجێکردنی شێوەی پێکهاتنی زانیاری پێویستی بە زۆرترین کات Maximum Time هەبێت، ئەوە خراپترین دۆخە. واتە زۆرترین کات داواکراو و پێویستە بۆ جێبەجێکردنی بەرنامەکە.
- دۆخی تێکرایی Average Case: لەو دۆخەدا، تێکرایی کات Average Time داواکراوو و، پێویستە بۆ جێبەجێکردنی بەرنامەکە.
- دۆخی باشترین Best Case: لەم دۆخەدا، کەمترین کات Minimum Time داواکراوو و، پێویستە بۆ جێبەجێکردنی بەرنامەکە.

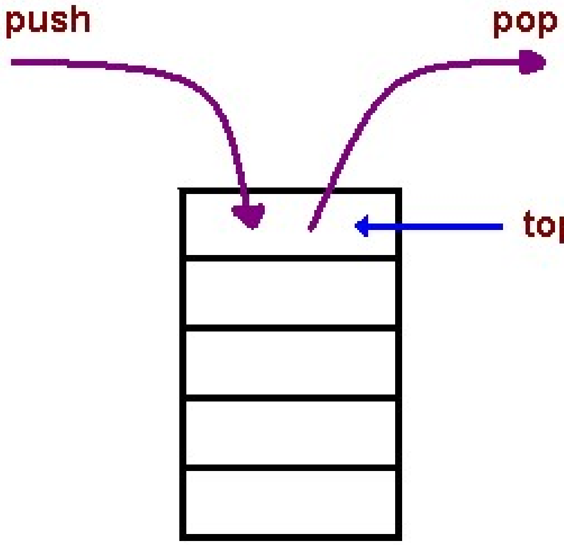
کردارە بنچینەییەکانی شێوەی پێکهاتنی زانیاری

Basic Operations of Data Structure

هەموو زانیارییی لە ناو شێوەی پێکهاتنی زانیاری دا، کردارەکانی لەسەر جێبەجێدەکرێت. کە بە گوێرەی جۆری زانیارییەکە ، و شێوەی پێکهاتنی زانیارییەکە، کردارەکان جیاوازن:

- تێپەڕین Traversing.

- گەران Searching.
- خستنه ناو ((تێخستن)) Insertion.
- سڕینەوە Deletion.
- ریزکردن ((نۆبە پێدان)) Sorting.
- یەکپێگرتن ((بەیەککردن)) Merging.
- نوێکردنەوە Updating.



بەشی دووەم

ڕیزکراوەکان

Arrays

لیستەیی

پێکەوەبەستراوو

Linked List

نووسینی: هێمن مەلا کەریم بەرزنجی

لیستەی پێکەوەبەستراوو Linked List

لیستەی پێکەوە بەستراوو Linked List شێوەی پێکهاتنی زانیاریی ڕێك و هێڵییە Linear Data Structure، کە زنجیرەیەکن لە پێکهاتنی زانیاری Sequence of Data Structure، و لە ڕێگەی بەستەرەکانەوە Links پێکەوەبەستراونەتەوە و بەیەك گەیەنراون.

لیستەی پێکەوەبەستراو Linked List زنجیرەیەك عەلقەیە Links کە دانەکان و پێکهاتەکانی Items تیایە، هەر عەلقەیەکیش Link پێکەوەبەستنی هەیە لەگەڵ عەڵقەیەکی تر Link، لیستەی پێکەوە بەستراو، لە دووای ئەرەیی Array دووەم جۆرە کە زۆر بەکاردێت، ئەم زاراوانەی خواراوە، زۆر گرنگن بۆ تێگەیشتن لە چەمکەکانی ((لیستەی پێکەوەبەستراوو Linked List)):

1-عەلقە Link: هەر عەلقەیەك لە لیستەی پێکەوەبەستراوو Linked Listدا، زانیاری Data تیاهەڵدەگیرێت و پاشەکەوت دەکرێت، کە پێی دەوترێت بەشی سەرەکی ((پێکهاتە/دانە)) Element.

2-دوواتر Next: هەر عەڵقەیەك Link لە لیستەی پێکەوەبەستراوو Linked Listدا، بەستەرێکی هەیە Link بۆ عەڵقەی دوواتر کە پێی دەڵێن: دوواتر Next.

3-لیستەی پێکەوەبەستراوو Linked List: عەڵقەی Link گرێدراوی تیادایە، کە بەستراوە بە عەڵقەی یەکەمەوە.

یاخود دەتوانین بڵێین: جۆرێکی زۆر باوی شێوەی پێکهاتنی زانیارییە کە پێکدێت لە زنجیرەیەك گرێ Node و، هەر گرێیەك دابەش بووە بۆ دوو بەش Two Parts، هەر گرێیەك Node پێکدێت لە زانیارییەکی خاوەنی Own Data و ناونیشانێك بۆ گرێی داهاتوو Address to Next Node، ئەم جۆرەی شێوەی پێکهاتنی زانیاری لە شێوەیی زنجیر و عەلقەیی پێکەوە بەستراو دایە. لیستەی پێکەوەبەستراوو Linked List بەکاردەهێنرێت بۆ بۆ دروستکردنی درەخت Tree و گراف Graph.

سودەكانی لیستەی پێكەوەبەستراوو

Advantage of Linked List

- سروشتێكی داینەمیكی هەیە Dynamic Nature.
- كردارەكانی تیادانان ((خستنە ناو)) Insertion و سڕینەوە ((لابردن)) Deletion بە ئاسانی جێبەجێدەكرێت Easily Implement.
- لیستەی پێكەوەبەستراو Linked List كاتی دەستپێگەیشتن Access Time كەمدەكاتەوە.

كەم و كورتییەكانی لیستەی پێكەوە بەستراوو

Disadvantage of Linked List

- بەفیڕۆچوونی میمۆری Memory Waste، چونكە ئاماژەدەرەكان Pointers پێویستی بە میمۆری زیاترە بۆ پاشەكەوتكردن Storage.
- دەستپێگەیشتنی دانەكانی بەشێوەیەكی هەرەمەكییانە Randomly نییە، بەڵكو دەستگەیشتن Access بەهەر گرێیەك Node بەشێوەیەكی زنجیرەییە Sequentially.
- تێپەڕینی پێچەوانەیی Reverse Traversingقورسە لە لیستەی پێكەوەبەستراودا Linked List.

جێبەجێكردنەكانی لیستەی پێكەوەبەستراو

Applications Linked List

- لیستەی پێكەوەبەستراو Linked List بەكاردەهێنرێت بۆ جێبەجێكردنی ستاك Stack، كیوو Queue و گراف Graph و ......تاد.
- لیستەی پێكەوەبەستراو Linked List ڕێگەمان پێدەدات، دانەی پێكهێنەر Element كە گرێیە Node لەسەرەتاوە Beginning یان لەكۆتایی End لیستەوە List زیاد بكەین.

- له لیستەی پێکەوە بەستراو Linked Listدا، پێویست نییە پێشوەختە حەجم بزانین و دیاری بکەین.

پیشاندانی لیستەی پێکەوە بەستراوو

Linked List Representation

لیستەر پێکەوەبەستراوو Linked List بە شێوەی وێنە دەخرێتە ڕوو لە زنجیرەیەك Sequence گرێدا Nodes، کە هەر گرێیەك ئاماژەدەکات بۆ گرێی دوواتر Next Node.

ئەم وێنەیە لیستەی پێکەوەبەستراوو Linked List ڕووندەکاتەوە و، پێویستە ئەم خاڵانەی لای خوارەوە ڕەچاو بکەین:

1-عەڵقەی یەکەم لە لیستەی پێکەوەبەستراوو دا، سەرە یە Head.

2-هەرعەڵقەیەك لە 2 بەش پێکهاتووە، بەشێکیان زانیارییە Data و، بەشەکەی تر بەستەرە Link بەناوی Next.

3-هەر عەڵقەیەك Link بە عەڵقەی دوواتر دەبەسترێتەوە بە هۆی بەستەری دوواتر Next Link.

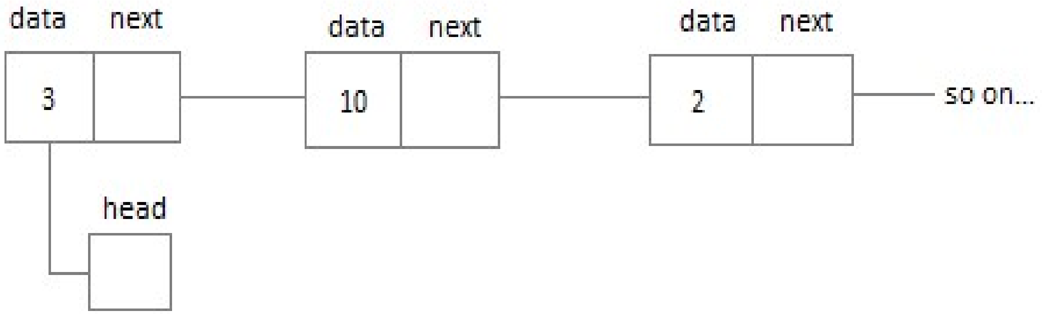
4-کۆتا عەڵقەش ((گرێش))، بەتاڵی Null هەڵگرتووە، کە ئاماژەیە بۆ کۆتایی لیستەکە.

جۆرەکانی لیستەی پێکەوەبەستراو

Types of Linked List

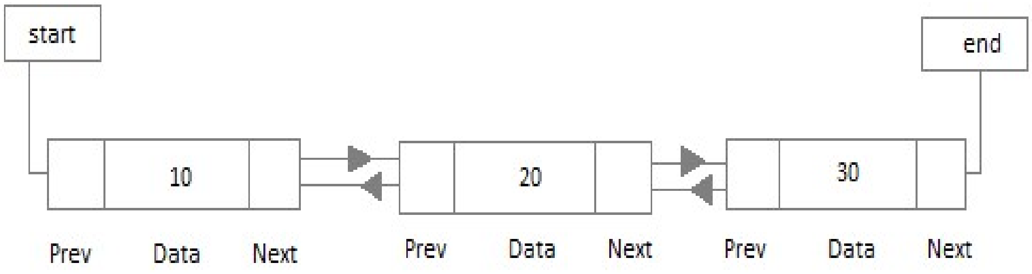
1-لیستەی پێکەوەبەستراویی تاك ئاڕاستە Singly Linked List:

لە جۆری لیستەی پێکەوەبەستراوی تاك ئاراستە دا Single Linked List، گرێکان Nodes هەن کە بەشی زانیارییان هەیە Data Part و، هەروەها بەشی ناونیشانیشان هەیە Address Part، واتە داهاتوو Next، کە ئاماژەدەکات Point بۆ گرێی داهاتوو Next Node لە زنجیرە گرێیەکدا Sequence of Nodes، ئەو کردارانەی لەسەر لیستەی پێکەوەبەستراوی تاك ئاراستەیی Singly Linked List جیبەجێدەبێت، بریتین لە تیادان ((خستنەناو، تێخستن)) Insertion و، سڕینەوە ((لابردن)) Deletion و، بڕینی هاتوچۆ ((تێپەڕین)) Traversal.



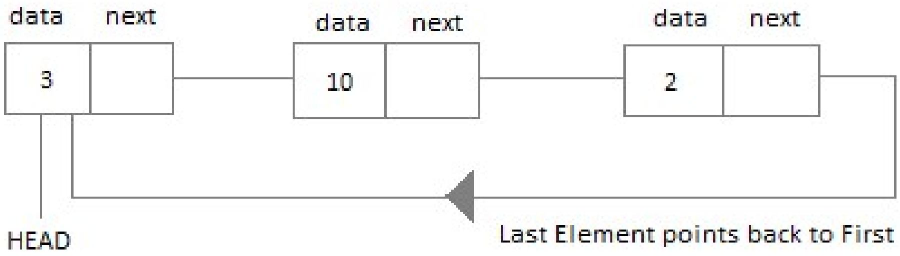
2-لیستەی پێکەوەبەستراویی دوو ئاراستە Doubly Linked List:

لەم جۆرەی لیستەی پێکەوەبەستراودا Doubly Linked List دوو بەستەر Link هەیە لەهەر نۆدێکدا، واتە هەر گرێیەك و دوو ناونیشانی Address هەیە، یەکەم بەستەر First Link ئاماژەدەکات بۆ گرێی پێشەوە Previous Node، و بەستەرەکەی تر Other Link ئاماژەدەکات بۆ گرێی دوواتر Next Node لە زنجیرەیەکدا Sequence.



3-لیستەی پێکەوەبەستراویی بازنەیی Circular Linked List:

لە جۆریی لیستەی پێکەوەبەسترانی بازنەیی Circular Linked Listدا، گرێی کۆتایی Last Node، ی لیستەکە، بەشی ناونیشان Address هەیە کە بۆیەکەم گرێ گەراوەتەوە و پێکەوەیی بەستوونەوە، ئەمەش شێوەی زنجیرەی بازنەی بەخشیوە بە شێوەی پێکهاتنەکە.



کردارە بنچینەییەکان

Basic Operation

ئەمانەی خوارەوە هەموو ئەو کردارە بنچینەییانەن کە لیستەی پێکەوەبەستراوو قبووڵی دەکات و پاڵپشتی دەکات:

-دانان ((تیادانانن/خستنە ناو)) Insertion: زیادکردنی یەك دانە لە سەرەتای لیستەوە.

-سرێنەوە Deletion: سڕینەوەی یەك دانە، لە سەرەتای لیستەوە.

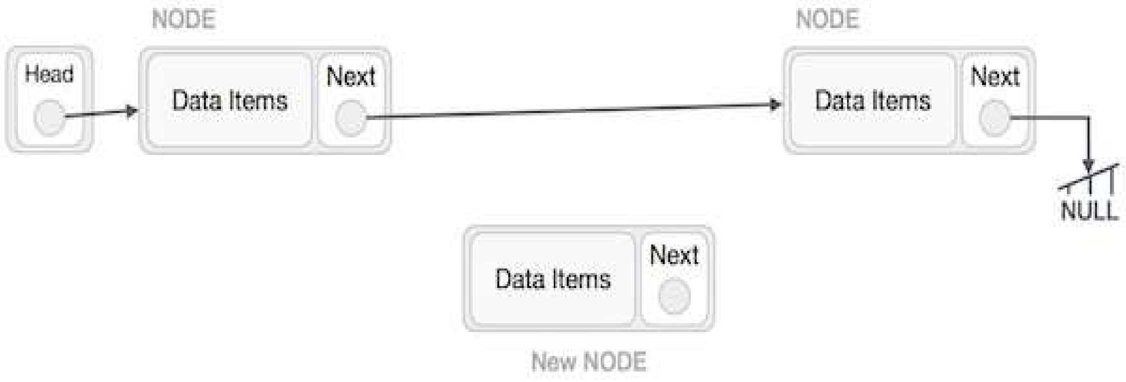
-پیشاندان Display: پیشاندانی هەموو لیستەکە.

-گەڕان Search: گەران بە شوێن دانەیەکی دیاری کراو بە بەکارهێنانی کلیلێکی دیاری کراو.

-سرینەوە Delete: سرینەوەی دانەیەکی دیاری کراو بە بەکارهێنانی کلیلێکی دیاری کراو.

کرداری خستنە ناو Insertion Operation

زیادکردنی گرێیەك Node لە لیستەی پێکەوە بەستراوو Linked Listدا، پێویستی بە زیاتر لە هەنگاوێکە، تا ئەنجام بدرێت، کە لە ڕێگەی ئەم وێنە هێلکارییانەوە ڕوونی دەکەمەوە، سەرەتا و یەکەم جار گرێیەك Node درووست دەکەین بە هەمان پێکهاتە، پاشان ئەو شوێنە دیاری دەکەین و دەیدۆزینەوە کە دەمانەوێت گرێ نوێیەکەی لێوە بخەینە ناو لیستە پێکەوەبەستراوەکە.

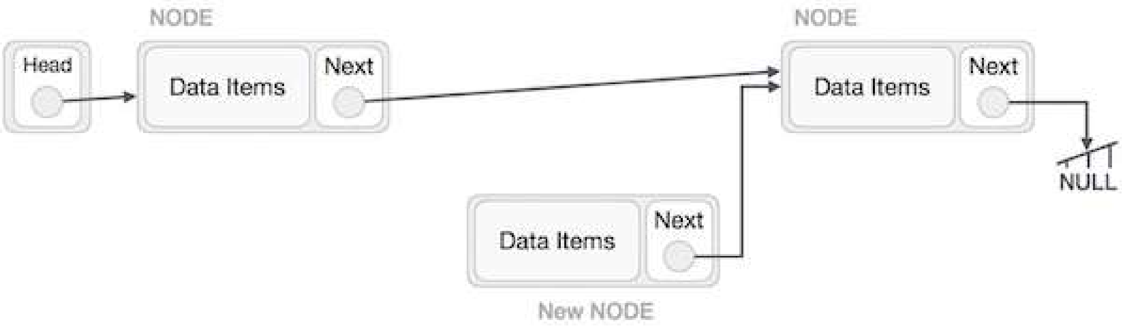


وا خەیاڵبکە، گرێی بی Node B کە گرێ نوێیەکەیە New Node بخەینە ناو لیستەکەوە لە نێوان ئەی A کە گرێی لای چەپە Left Node، و سی C کە گرێی لای ڕاستە Right Node.

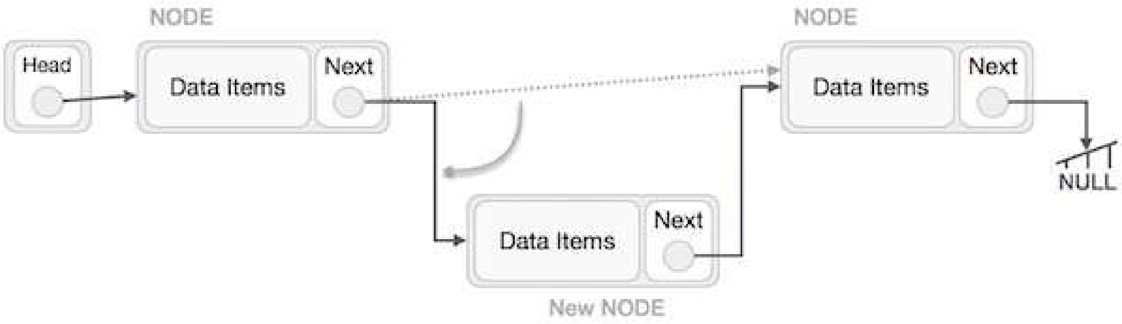
پاشان :

```
NewNode.next -> RightNode;
```

بەم جۆرەی لای خوارەوە، ڕوو دەدات:



ئێستا پێویستە ((دوواتر Next)) ی گرێی لای چەپ Left Node بچێت بۆ گرێ نوێیەکە New Node:



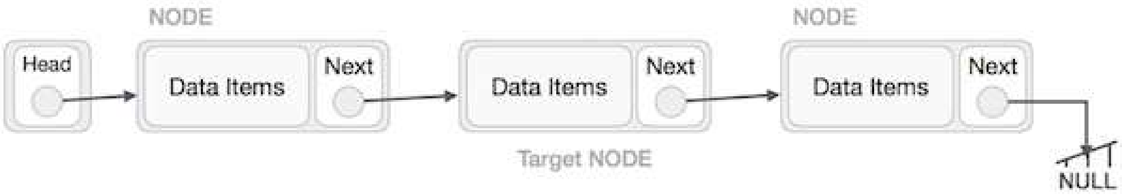
بەم جۆرە گرێیەکمان Node خستە نێوان دوو گرێوە:

بەهەمان شێوە و رێگە دەتوانین، گرێ زیاد بکەین لە هەر شوێنێك کە بمانەوێت.

کرداری سڕینەوە

Delete Operation

سڕینەوەش Deletion زیاتر لە هەنگاوێکی پێویستە، کە من بەهۆی وێنەوە هەنگاوەکان دەخەمە ڕوو، ڕوونی دەکەمەوە، یەکەم جار شوێنی ئەو گرێیە Node دیاری دەکەین کە دەمانەوێت لایبەرین Remove، بە هۆی ئەلگۆریسمی گەڕانەوە Searching Algorithm:

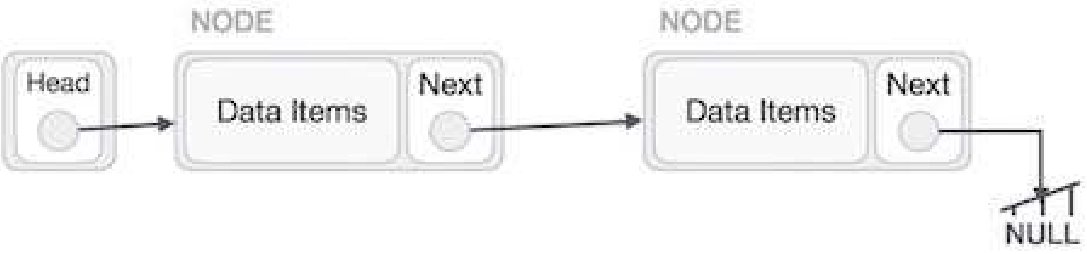


گرێی Node پێشەوە Previous و لای چەپ Left Node ئەو گرێیەی Node دەمانەوێت لای بەرین، ئاماژەپێدەکەین Point و دەیبەین بۆ گرێی دووای ئەو گرێیەی دەمانەوێت لای بەرین.

ئەو بەستەرە Link لادەبەین Remove کە چووە بۆ ئەو گرێیەی دەمانەوێت بیسرینەوە، ئێستا دەتوانین ئەو گرێیە Node بسرینەوە کە مەبەستمان بوو.

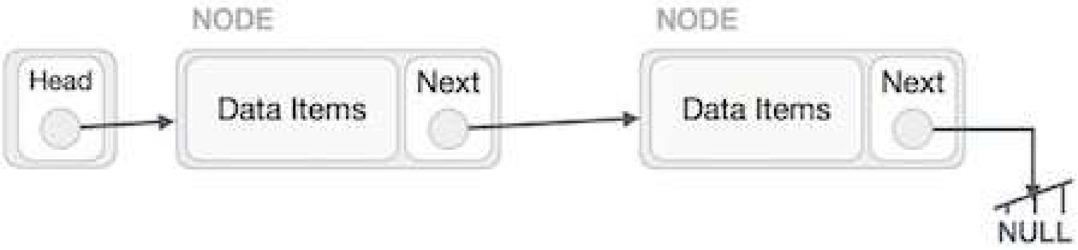
```
TargetNode.next -> NULL;
```

پێویستە سوود وەربگرین لە کرداری سرینەوەی گرێ Delete Node، چونکە لە میمۆری Memoryدا، دەمێنێتەوە، ئەگەر ئەو شوێنەی لێوەرنەگیرێتەوە Deallocate کە لە میمۆری Memoryدا، بۆی دیاری کراوە.

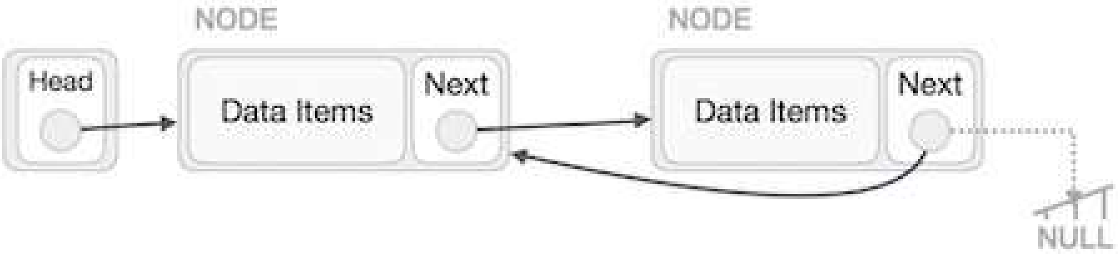


کرداری پێپێچەوانەیی Reverse Operation

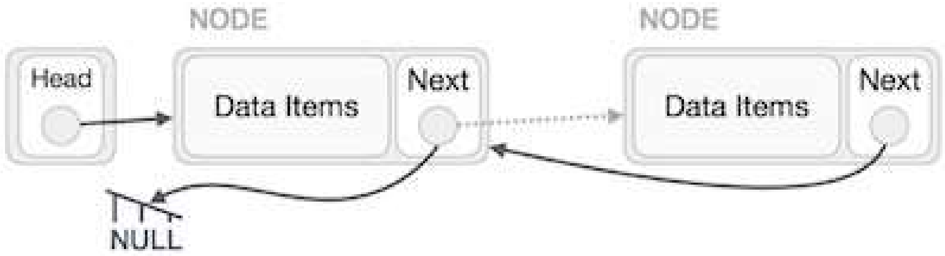
لەم کردارەدا، پێویستە گرێی کۆتایی End Node بچێت بۆ گرێی سەرە Head Node و، پێچەوانەکردنەوەی هەموو لیستە پێکەوەبەستراوەکە Linked List.



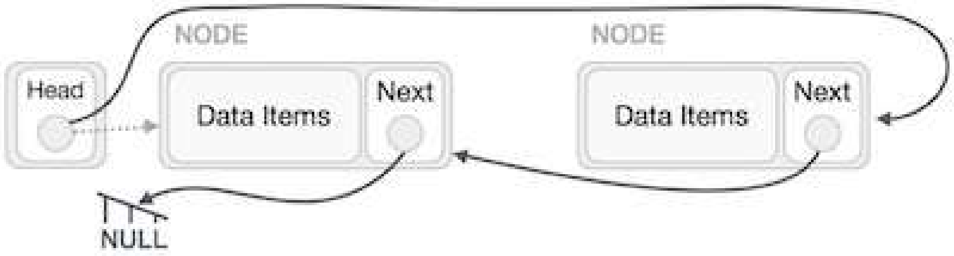
یەکەمجار، پێویستە کۆتایی لیستەکە پێچەوانە بکەینەوە، کە چووە بۆ بەتاڵ Null، ئێستا پێویستە ئەم گرێیە بەرین بۆ گرێی پێشوو.



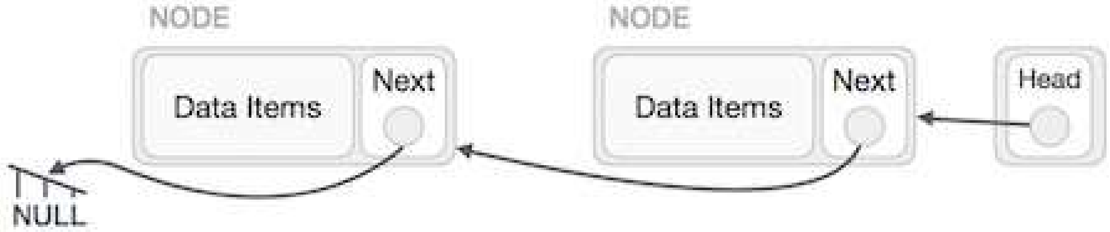
پێویستە لەوە دڵنیابین کە گرێی کۆتایی، گرێیەکی بەفیڕۆچوو Lost Node نییە، پاشان گرێی لای چەپ ببەین بۆ بەتاڵ Null.



پاشان گرێی سەرە Head Node دەبەین بۆ گرێی کۆتایی و ئاماژەی پێدەکەین، بەمەش گرێی یەکەم ئاماژەدەکات و دەچێت بۆ بەتاڵ Null.



ئێستا بەستەری سەرە Head Link کە چووبوو بۆ گرێی یەکەم لادەبەین و، خۆشمان بردمان بۆ گرێی کۆتایی، بەمەش پێچەوانە بوویەوە و گرێی کۆتایی بوو بە گرێی یەکەم.



لیستەی ڕێکخراو Linked List پێچەوانە بوویەوە و، وەك دەبینن کارێکی قورس و ئاڵۆز بوو، بۆیە پێشتر وتمان کە کرداری پێچەوانەکردنەوە قورسە Difficult.

لیستەی پێکەوەبەستراویی دوو هێندە

Doubly Linked List

ئەم جۆرەی لیستەی پێکەوە بەستراو Linked List، گونجاوە بۆ ئەوەی بە هەردوو ڕێگەیی بۆ پێشەوە و بۆ دووارەوە Forward and Backward، بڕووات بە بەراورد لەگەڵ ((لیستەی پێکەوە بەستراویی تاك و سادە Single Linked List)).

ئەم زاراوانەی لای خوارەوە زۆر پێویستن بۆ تێگەیشتن لە بیرۆکە و چەمکەکانی ((لیستەی پێکەوە بەستراوویی دوو هێندە Doubly Linked List)):

1-عەلقە Link: هەر عەلقەیەك لە لیستەی پێکەوەبەستراوو Linked Listدا، زانیاری Data تیاهەڵدەگیرێت و پاشەکەوت دەکرێت، کە پێی دەوترێت بەشی سەرەکی ((پێکهاتە/دانە)) Element.

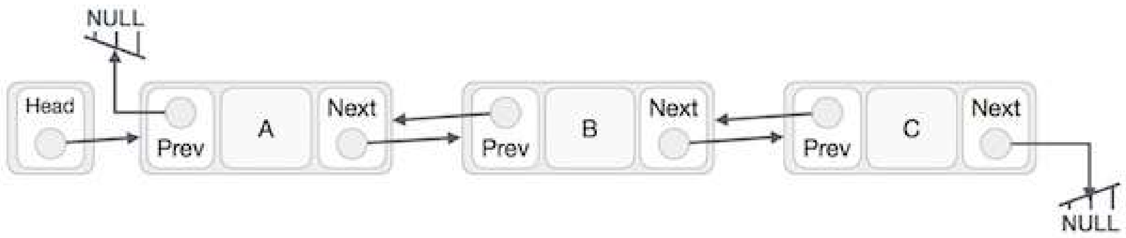
2-دوواتر Next: هەر عەلقەیەك Link لە لیستەی پێکەوەبەستراوو Linked Listدا، بەستەرێکی هەیە Link بۆ عەلقەی دوواتر کە پێی دەلێن: دوواتر Next.

3-پێشەوە Prev: هەر عەلقەیەك Link لە لیستەی پێکەوەبەستراوو Linked Listدا، بەستەرێکی هەیە Link بۆ عەلقەی پێشەوە ((پێشتر)) Previous کە پێی دەلێن: پێشتر Prev.

4-لیستەی پێکەوەبەستراوو Linked List: عەلقەی Link گرێدراوی تیادایە، کە بەستراوە بە عەلقەی یەکەمەوە ، و پێی دەلێن: یەکەم First، هەروەها بۆ عەلەی کۆتایی Last Link کە پێی دەلێن: کۆتایی Last.

پیشاندانی لیستەی پێکەوە بەستراویی دوو هێندە

Doubly Linked List Presentation



به سەرنج دان لە وێنەکە، ئەم خاڵانەی لای خوارەوە دەخەمە ڕوو:

- لە لیستەی پێکەوە بەستراوی دوو هێندەدا، عەڵقەیی سەرەتایی و کۆتایمان هەیە، کە پێی دەلێن: یەکەم First و کۆتا Last.
- هەر عەڵقەیەك Link بەشی زانیاری Data هەیە، لەگەڵ بەشی بەستەر Link کە پێی دەلێن : داهاتوو Next.
- هەر عەڵقەیەك Link بەستراوە بە عەلەی دوواترەوە Next Link لە ڕێگەی بەستەری دوواترەوە Next Link.
- هەر عەلەقەیك Link، بە ستراوە بە عەلقەیی پێشخۆیەوە Previous Link لە ڕێگەی بەستەری پێشتر Prev Link.
- عەڵقەیی کۆتایی Last Link، عەڵقەیەکی بەتاڵە Null بۆ دیاری کردنی کۆتایی لیست.

کردارە بنچینەییەکان

Basic Operations

ئەم کردارانەی لای خوارەوە هەموو ئەو کردارە بنچینەییانەن کە لە لایەن لیستەی پێکەوە بەستراویی دوو هێندەوە Doubly Linked List قبووڵدەکرێت و، ئەنجام دەدرێت:

-دانان ((تیادانانن/خستنە ناو)) Insertion: زیادکردنی یەك دانە لە سەرەتای لیستەوە.

-سرێنەوە Deletion: سڕینەوەی یەك دانە ، لە سەرەتای لیستەوە.

-دانان ((تیادانانن/خستنە ناو))لە کۆتاییەوە   Insert Last: زیادکردنی یەك دانە لە کۆتایی لیستەوە.

-سرێنەوە لە کۆتاییەوە Delete Last: سڕینەوەی یەك دانە ، لە کۆتایی لیستەوە.

-دانان ((تیادانانن/خستنە ناو))لە دووا   Insert After: زیادکردنی یەك دانە لە دووایی هەر دانەیەکی لیستەوە.

-پیشاندان بەرەو پێشەوە   Display Forward: پیشاندانی هەموو لیستەکە بەرەو پێشەوە و لە سەرەتاوە بۆ کۆتایی.

-پیشاندان بەرەو دوواوە   Display Backward: پیشاندانی هەموو لیستەکە بەرەو دوواوە و لە دوواوە بۆ پێشەوە.

-سرینەوە Delete: سرینەوەی دانەیەکی دیاری کراو بە بەکارهێنانی کلیلێکی دیاری کراو.

کرداری خستنە ناو Insertion Operation

ئەم کۆدەی لای خوارەوە، کرداری دانانی گرێیەکی نویێ، ڕووندەکاتەوە لە سەرەتایی ((لیستەی پێکەوە بەستراووی دوو هێندە Double Linked List)):

```
//insert link at the first location
void insertFirst(int key, int data) {

   //create a link
   struct node *link = (struct node*) malloc(sizeof(struct node));
   link->key = key;
   link->data = data;
```

```
    if(isEmpty()) {
       //make it the last link
       last = link;
    }else {
       //update first prev link
       head->prev = link;
    }

    //point it to old first link
    link->next = head;

    //point first to new first link
    head = link;
}
```

Deletion Operation کرداری سڕینەوە

ئەم کۆدەی لای خوارەوە، کرداری سرینەوەی گرێیەکی نوێ، ڕووندەکاتەوە لە سەرەتایی ((لیستەی پێکەوە بەستراووی دوو هێندە Double Linked List)):

```
//delete first item
struct node* deleteFirst() {

    //save reference to first link
    struct node *tempLink = head;

    //if only one link
    if(head->next == NULL) {
```

```
      last = NULL;
   }else {
```

```
      head->next->prev = NULL;
   }

   head = head->next;

   //return the deleted link
   return tempLink;
}
```

کرداری خستنە ناو لە کۆتاییەوە Insertion at End Operation

ئەم کۆدەی لای خوارەوە، کرداری دانانی گرێیەکی نوێ، ڕوونده‌کاته‌وه لە کۆتایی ((لیستەی پێکەوە بەستراووی دوو هێندە Double Linked List)):

```
//insert link at the last location
void insertLast(int key, int data) {

   //create a link
   struct node *link = (struct node*) malloc(sizeof(struct
node));
   link->key = key;
   link->data = data;

   if(isEmpty()) {
      //make it the last link
      last = link;
   }else {
```

```
      //make link a new last link
      last->next = link;
      //mark old last node as prev of new link
      link->prev = last;
   }

   //point last to new last node
   last = link;
}
```

لیستەی پێکەوەبەستراویی بازنەیی

Circular Linked List

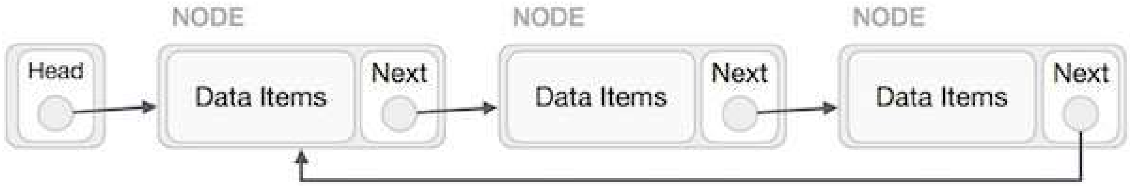
لەم جۆرەی شێوەی پێکهاتنی زانیاری Data Structure Type دا، یەکەم دانە First Element دەچێت بۆ کۆتا دانە Last Element و، کۆتا دانەش Last Element دێتەوە بۆ یەکەم دانە First Element.

هەردوو ((لیستەی پێکەوە بەستراووی تاك ئاراستە Single Linked List)) و ((لیستەی پێکەوە بەستراووی دوو ئاراستە Doubly Linked List)) بەکاردەهێنرێن لە درووستکردنی ((لیستەی پێکەوەبەستراویی بازنەیی Circular Linked List)) دا.

لیستەی پێکەوەبەستراوویی تاك ئاراستە وەکو بازنەیی

Singly Linked List as Circular

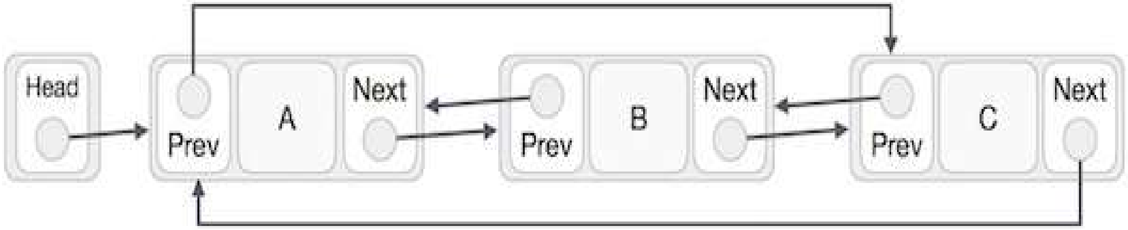
لە لیستەی تاك ئاراستە Single Linked Listدا، ئاماژەدەری دوواتر Next Pointer ی کۆتا گرێ Last Node دەگەرێتەوە بۆ یەکەم گرێ First Node و، بەمەش شێوەی بازنەیی Circular درووست دەبێت.



لیستەی پێکەوەبەستراوویی دوو ئاراستە وەکو بازنەیی

Doubly Linked List as Circular

لە لیستەی پێکەوەبەستراوویی دوو ئاراستە Doubly Linked List دا، ئاماژەدەری دوواتر Next Pointer ی کۆتا گرێ Last Node دەگەرێتەوە و ئاماژەدەکات بۆ یەکەم گرێ First Node، هەروەها ئاماژە دەری پێشتر Previous Pointer ی یەکەم گرێ First Nodeدەژێت بۆ ئاماژەدەری پێشتری Previous Node کۆتا گرێ Last Node ،



کردارە بنچینەییەکان

Basic Operations

ئەم کردارانەی لای خوارەوە لە گرنگترین کردارەکانن کە پالْپشتی دەکرێن و، قبوولْدەکرێن لە لایەن لیستەی پێکەوەبەستراوویی بازنەییەوە Circular Linked List:

-دانان ((تیادانانن/خستنە ناو)) Insertion: زیادکردنی یەك دانە لە سەرەتای لیستەوە.

-سرێنەوە Deletion: سڕینەوەی یەك دانە، لە سەرەتای لیستەوە.

-پیشاندان Display: پیشاندانی لیستەکە.

کرداری خستنە ناو Insertion Operation

ئەم کۆدەی لای خوارەوە، کرداری دانانی گرێیەکی نوێ، ڕووندەکاتەوە لە ((لیستەی پێکەوە بەستراووی بازنەیی Circular Linked List))دا، لەسەر بنچینەیی ((لیستەی پێکەوە بەستراوویی یەك ئاراستە Single Linked List)).

```
//insert link at the first location
void insertFirst(int key, int data) {
   //create a link
   struct node *link = (struct node*) malloc(sizeof(struct
node));
   link->key = key;
   link->data= data;

   if (isEmpty()) {
      head = link;
      head->next = head;
   }else {
      //point it to old first node
      link->next = head;

      //point first to new first node
      head = link;
   }

}
```

Deletion Operation کرداری سڕینەوە

ئەم کۆدەی لای خوارەوە، کرداری سرینەوەی گرێیەکی نوێ، ڕووندەکاتەوە لە ((لیستەی پێکەوە بەستراووی بازنەیی Circular Linked List))دا، لەسەر بنچینەیی ((لیستەی پێکەوەبەستراوویی یەك ئاراستە Single Linked List)).

```
//delete first item
struct node * deleteFirst() {
   //save reference to first link
   struct node *tempLink = head;

   if(head->next == head){
      head = NULL;
      return tempLink;
   }

   //mark next to first link as first
   head = head->next;

   //return the deleted link
   return tempLink;
}
```

کرداریی پیشاندان

Display Operation

ئەم کۆدەی لای خوارەوە کرداریی پیشاندان Display Operation ڕووندەکاتەوە لە ((لیستەی پێکەوەبەستراوویی بازنەیی Circular Linked List)) دا:

```
//display the list
void printList() {
   struct node *ptr = head;
   printf("\n[ ");

   //start from the beginning
   if(head != NULL) {
      while(ptr->next != ptr) {
         printf("(%d,%d) ",ptr->key,ptr->data);
         ptr = ptr->next;
      }
   }

   printf(" ]");
}
```

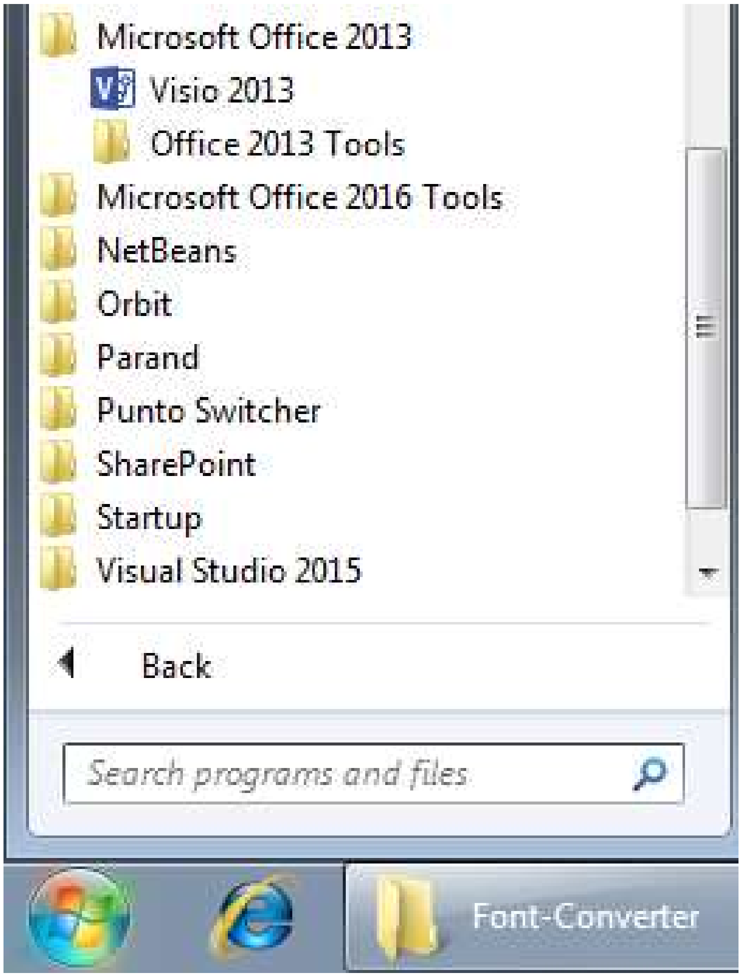
هێڵکاریی

شێوەی پێکهاتنی بەرنامە

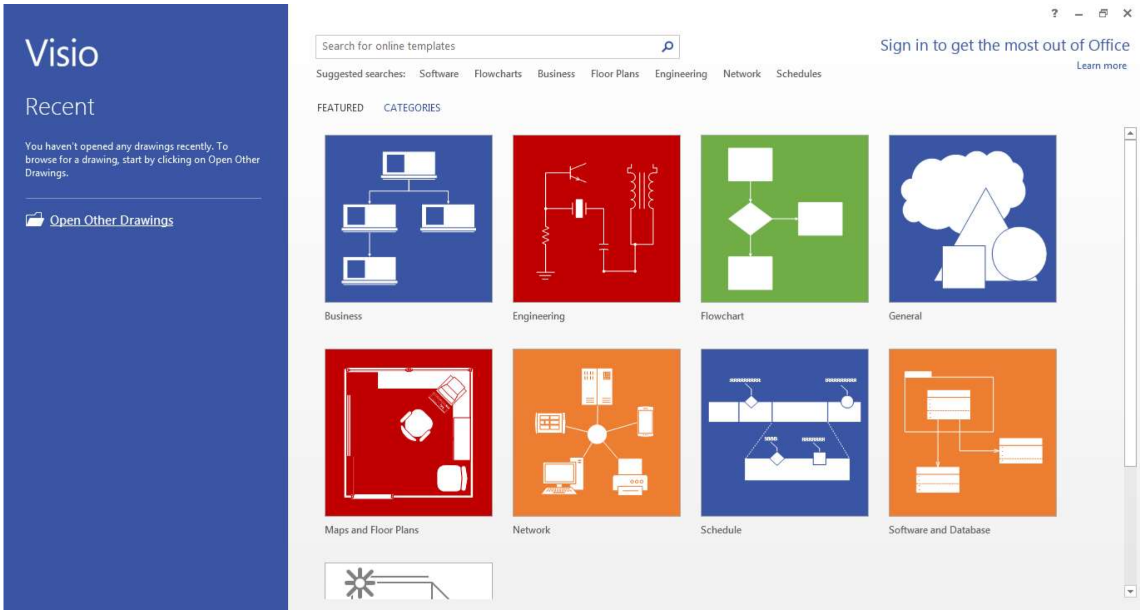
Program Structure Diagram

هەمیشە بەرنامەرێژیی و پێویستی دیاری کردن و نەخشە سازیی لە دیارترین هەنگاوەکانی کاری بوواری تەکنەلۆژین، یەکێك لەو هێڵکارییانەش کە بۆ وێنەکێشی ڕوونکردنەوەی شێوەی پێکهاتنی زانیاری و، خستنە ڕووی ((لیستەی پێکەوەبەستراوویی یەك ئاراستە Single Linked List)) و ((لیستەی پێکەوە بەستراوویی دوو ئاراستە Doubly Linked List)) و ((لیستەی پێکەوە بەستراوویی بازنەیی Circular Linked List)) بەکاردەهێنرێت، ناوی ((هێڵکاری شێوەی پێکهاتنی بەرنامەیە Program Structure)) کە لەبەشی بنکەی زانیاری و بەرنامە Software and Database بەرنامەی مایکرۆسۆفت ڤیزیۆ 2016 دانراوە، بۆیە بە کورتی باسی بەکارهێنانی دەکەین لە نموونەیەك دا.

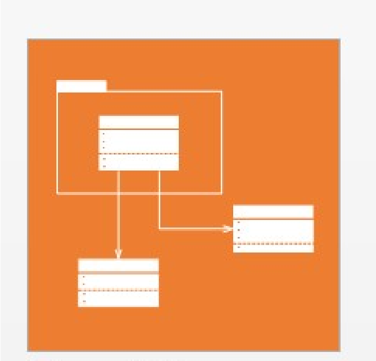
1-بەرنامەی مایکرۆسۆفت ڤیزیۆ 2013 دەکەینەوە، بە کلیك کردن لەسەر دووگمەی دەستپێکردن Start و، پاشان کلیك کردن لەسەر هەموو بەرنامەکان All Program، دوواتر کلیك کردن لەسەر مایکرۆسۆفت ئۆفیس 2013، لە کۆتایدا کلیك کردن لەسەر مایکرۆسۆت ڤیزیۆ 2013:



2-ئیستا بەرنامەکە کراوەتەوە:

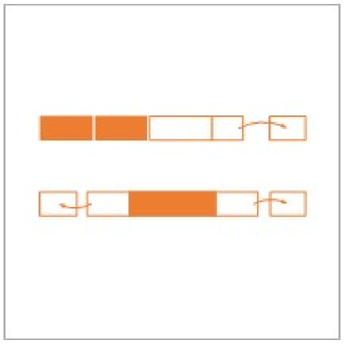


3- کلیك لەسەر پۆلێنی بنکەی زانیاری و بەرنامە Software and Database دەکەین:



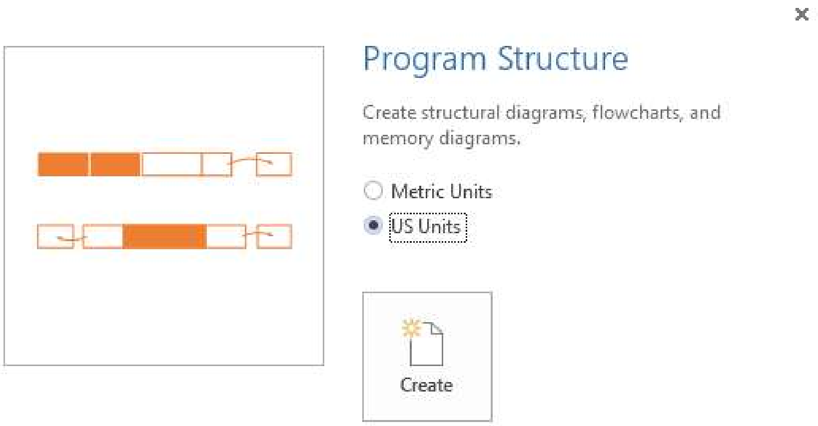
Software and Database

4-ئیستا بەشی بنکە ی زانیاری و بەرنامە کراوەتەوە و، کلیك لەسەر هێڵکاری شیوەی پێکهاتنی بەرنامە Program Structure Diagram، دەکەین:

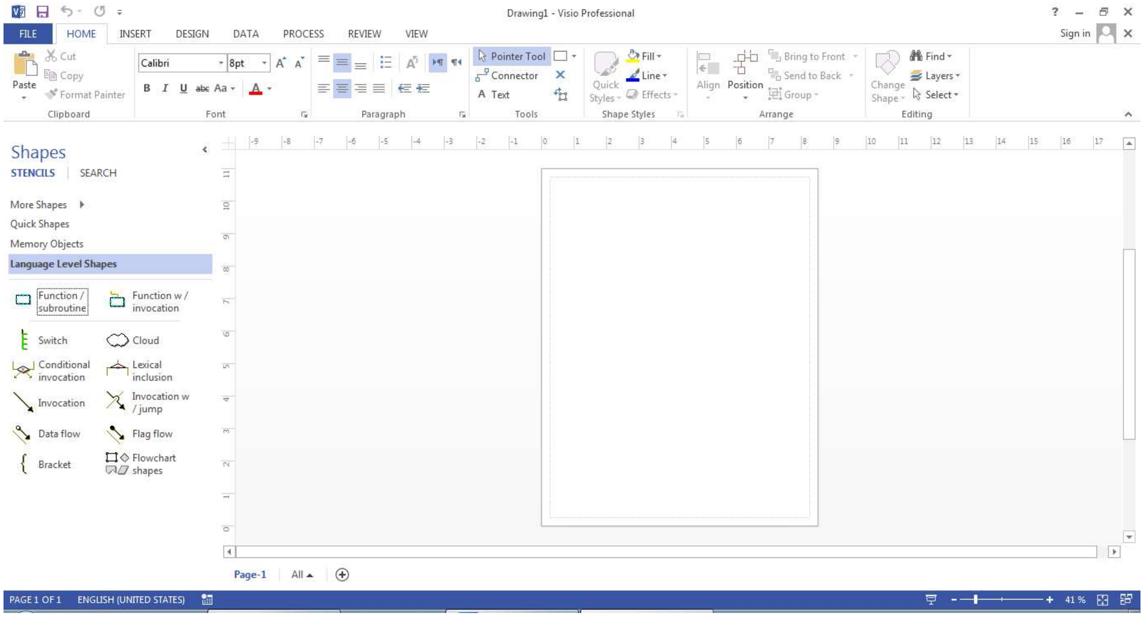


Program Structure

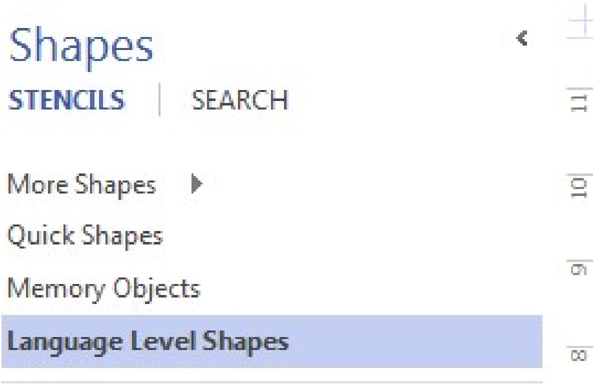
5-كليك لەسەر دووگمەی دروستکردن Createدەکەین:



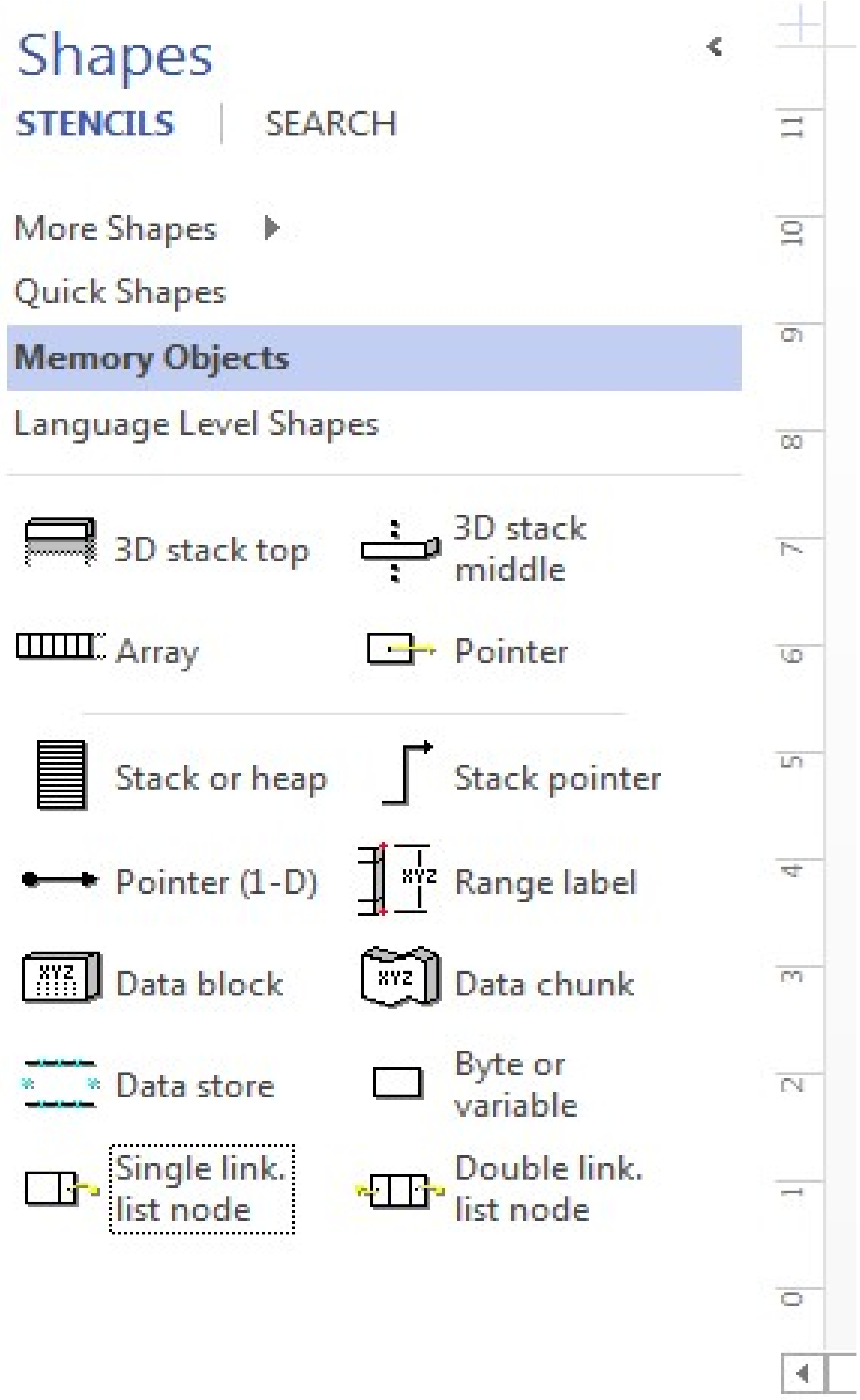
6-هێڵکارییەکە چالاکە بۆ کێشانی هەر جۆریکی لیستەی پێکەوە بەستراوو:



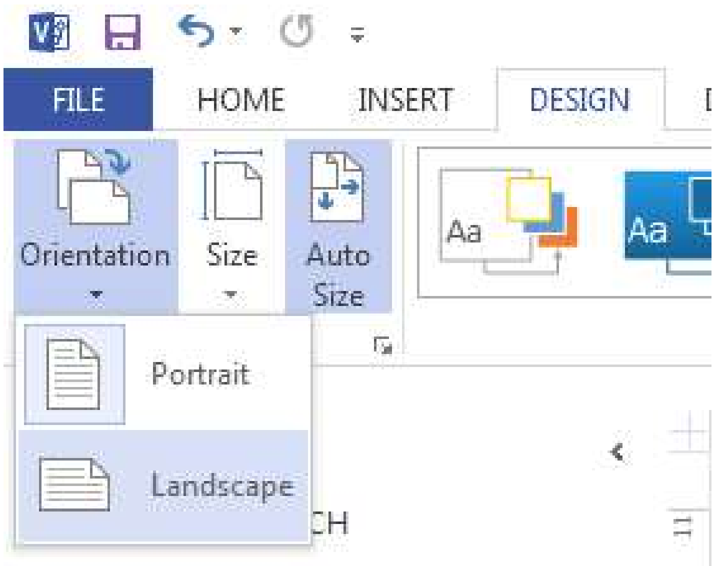
7-لە بەشی دەستەچەپی بەرنامەکە و لە ژێر شێوەکان Shapesدا، کلیك لەسەر تەنەکانی میمۆری Memory Objectsدەکەین:



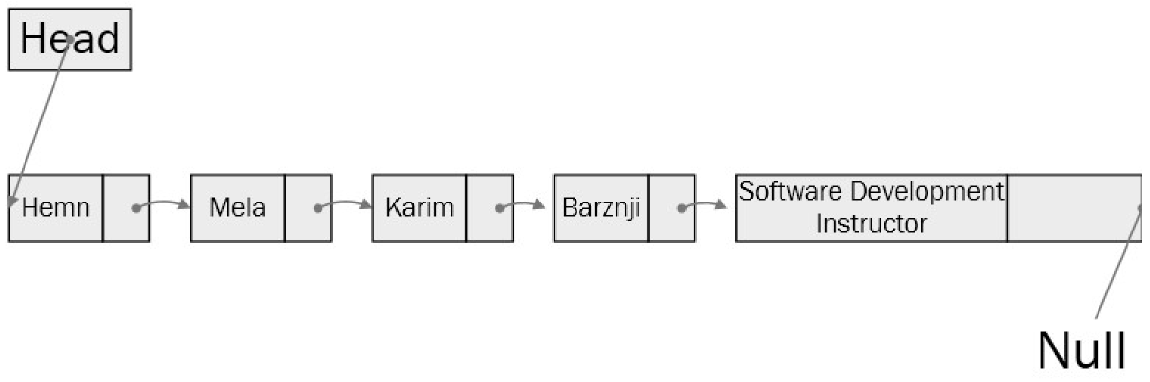
8-بەشەکانی تایبەت بە تەنەکانی میمۆری Memory Objects دەردەکەوێت و، هەر شێوەیەکان ویست هەڵیدەگرین و رای دەکێشینە سەر لاپەرەی کار :



9-ئەگەر هێڵکارییەکەمان لە باری درێژیی بێت ئەوا واباشترە لاپەڕەکە بخەینە باری درێژیی، بە کلیك کردن لە سەر تابی نەخشەسازیی Design، پاشان کلیك کردن لەسەر ئاراستە Orientation، و دوواتر کلیك کردن لەسەر Landscape:



10-ئێستا لیستەیەکی پێکەوە بەشتراوو دەکێشین، بۆیە هەریەکێك لە پێکهاتەکان هەڵەگرین و رایدەکێشین بۆ سەر لاپەڕەی کار و دەیان گەیەنین بە یەك هەر بە راکێشانی پێکهاتەکان:



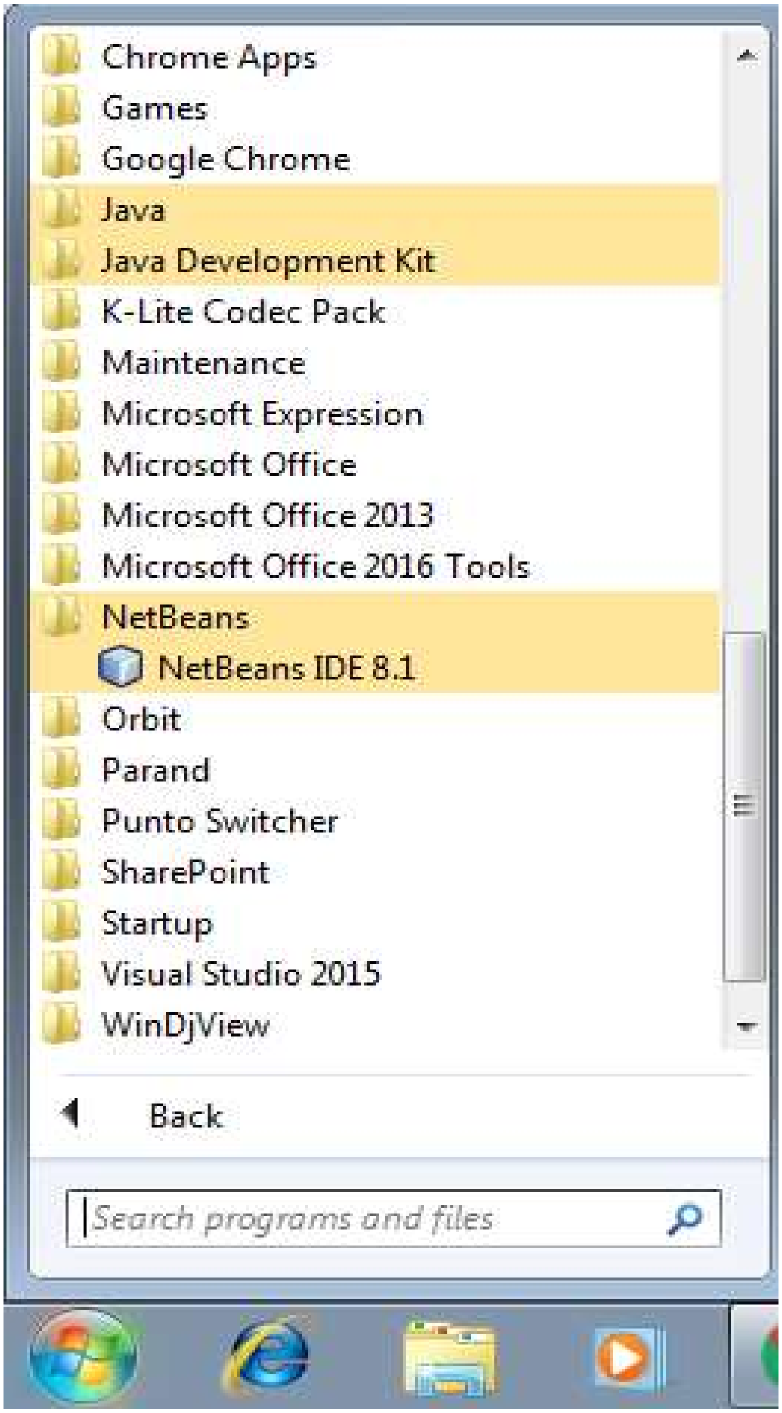
جێبەجێکردنی لیستەی پێکەوە بەستراوو

Linked List Implementation

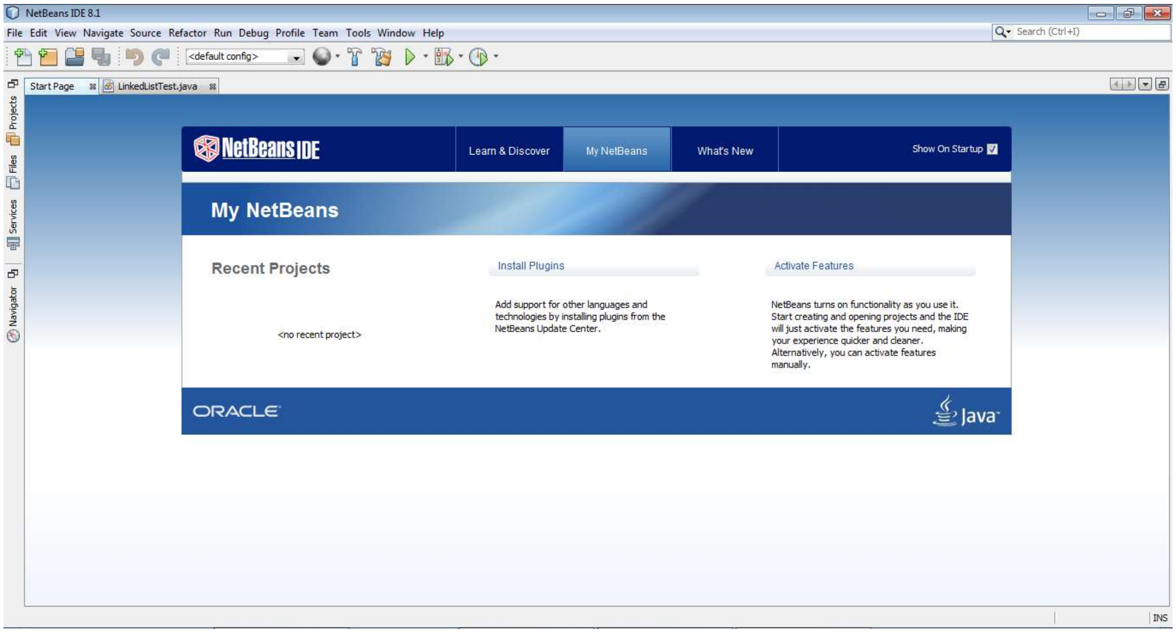
جێبەجێکردنی لیستەی پێکەوە بەستراوو

Linked List Implementation

1-بەرنامە نێتبین NetBeans دەکەینەوە، بە کلیك کردن لەسەر دووگمەی دەستپێکردن Start Button، پاشان کلیك لەسەر NetBeans و دوواتر کلیك لەسەر NetBeans IDE 8.1:

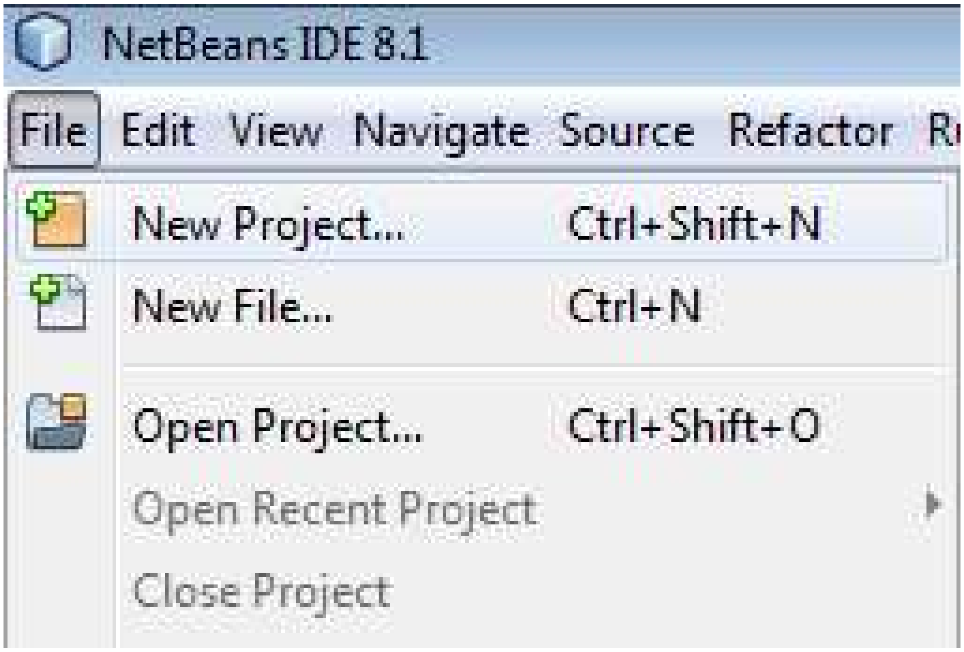


2-ڕووكاری بەرنامەی نێتبینز NetBeansدەكرێتەوە.

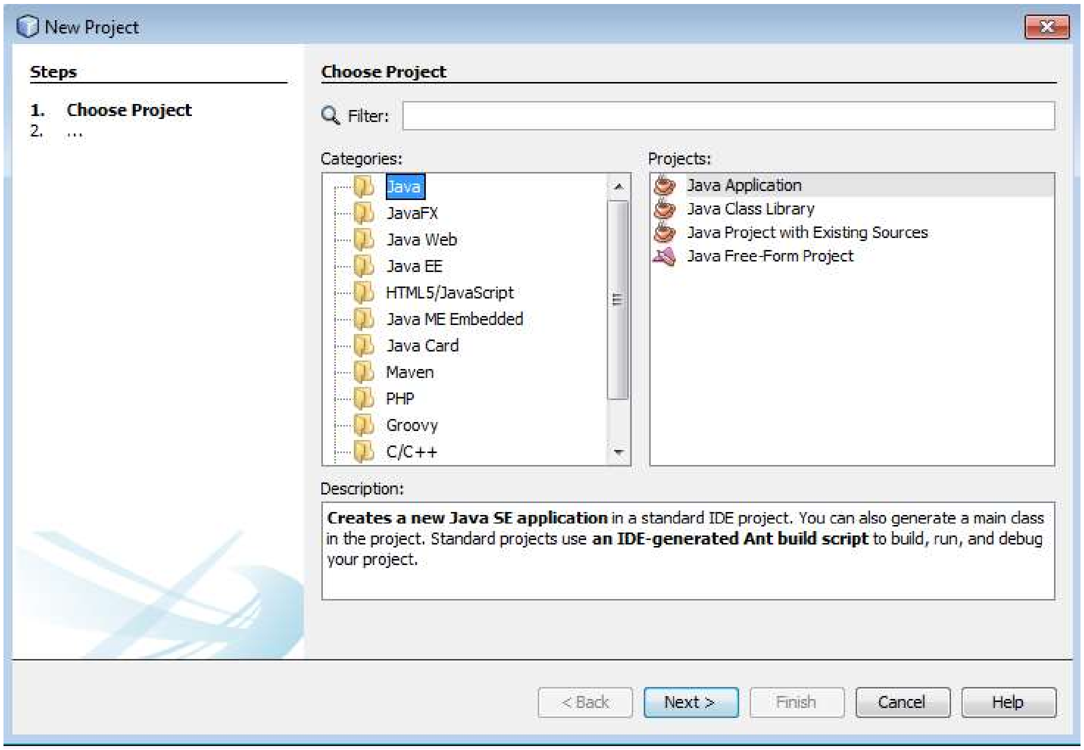


3-كلیك لەسەر دووگمەی فایل Fileدەكەین، بۆ ئەوەی پرۆژەیەكی نوێ درووست بكەین.

4-كلیك لەسەر پرۆژەی نوێ New Project دەكەین:



5-ڕووکاری تایبەت بە پرۆژەی نوێ New Projectدەکرێتەوە و، کلیك لەسەر دووگمەی دوواتر Next دەکەین.



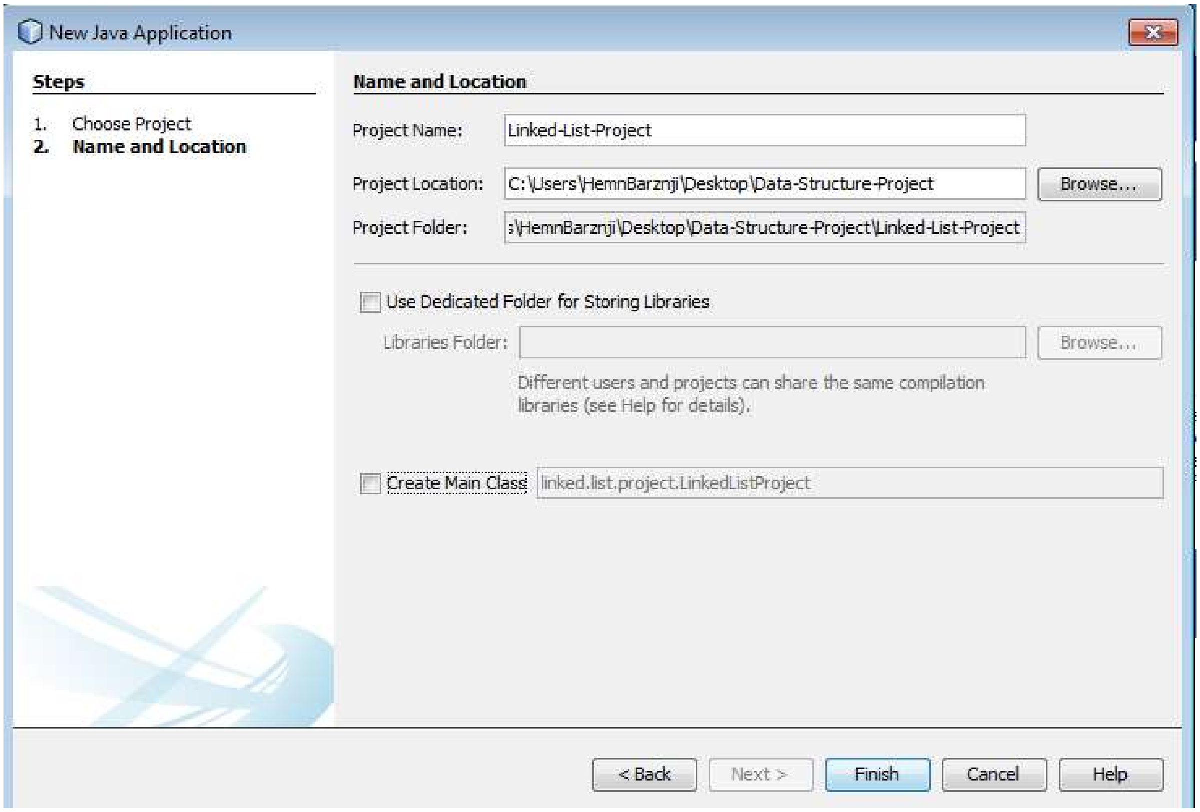
6-ڕووکارێکی نوێ دەکرێتەوە:

-لەبەشی ناوی پرۆژە Project Nameدا، ناوێك بۆ پرۆژەکە بنووسە.

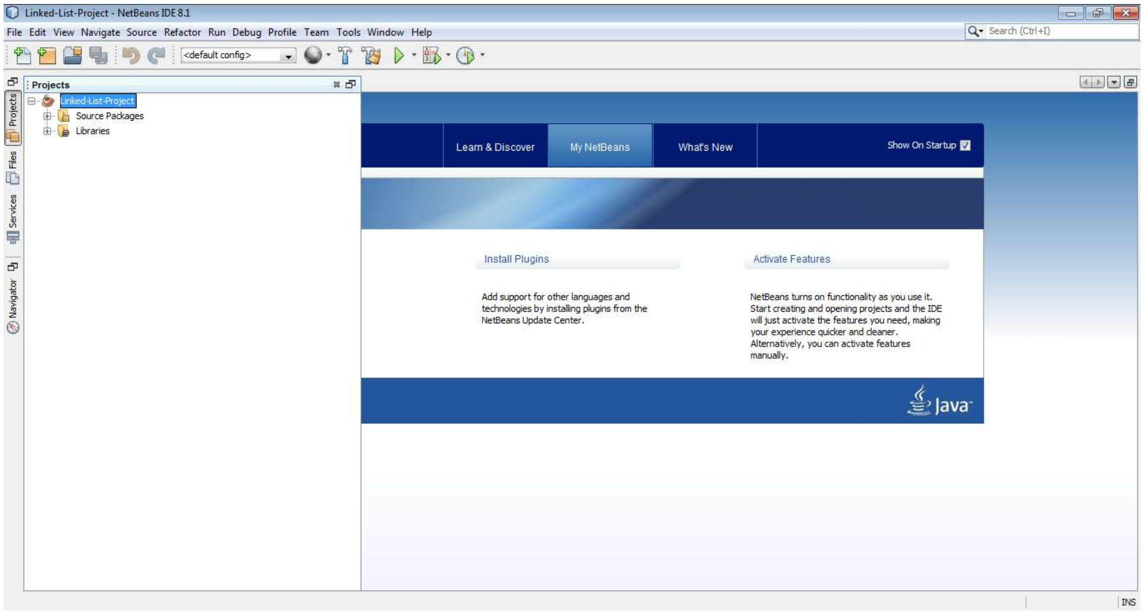
-لەبەشی شوێنی پرۆژە Project Locationدا، شوێنی پرۆژەکە دیاری بکە.

-کلیك لەناو سنووقی پشکنینی Check Box درووستکردنی پۆلی سەرەکی Create Main Class بکە، ئەگەر دەتەوێت پۆلێکت بۆ درووست بکات بە ناوێکەوە کە بەرنامەکە دیاری دەکات.

-لە دووای ئەو هەنگاوانە کلیك لەسەر دووگەی کۆتای هاتن Finishبکە.



7-ئیستا پرۆژەکە درووست بوو:



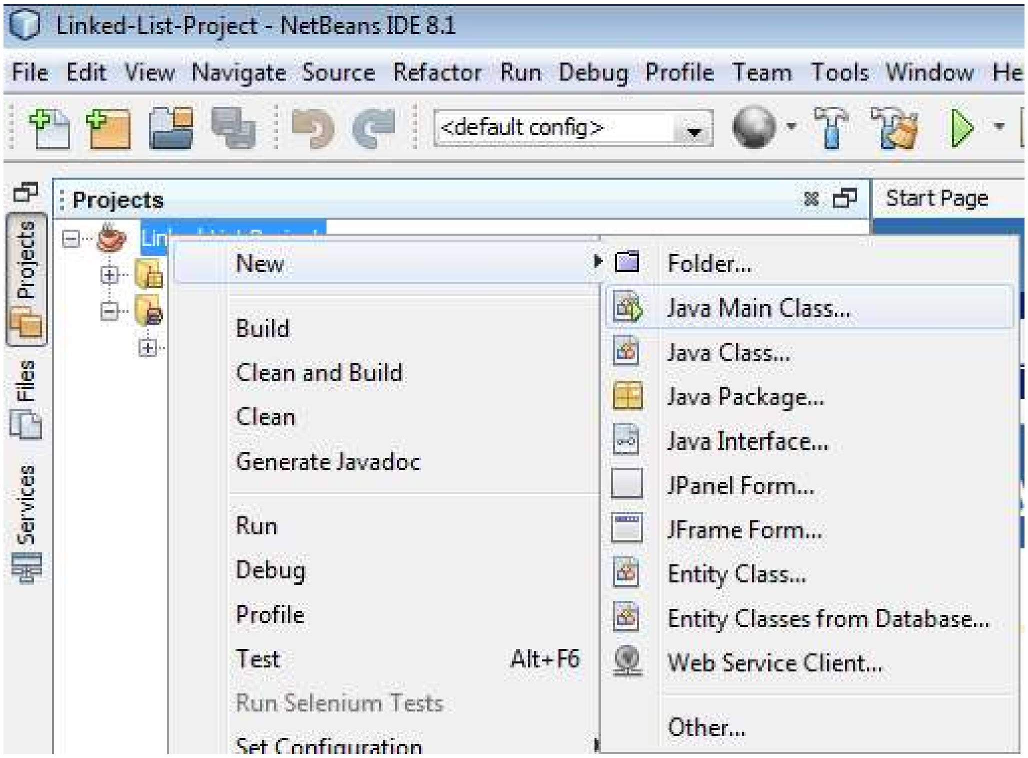
دروستکردنی پۆل

Create Class

1-لەلای چەپی بەرنامەکەوە بەشی پرۆژەکان Projects هەیە و لە سەر ناوی پرۆژەکە Project Name کلیکی راست Right Click بکە.

2-سەهمی ماوسەکە ببەرەسەر نوێ New.

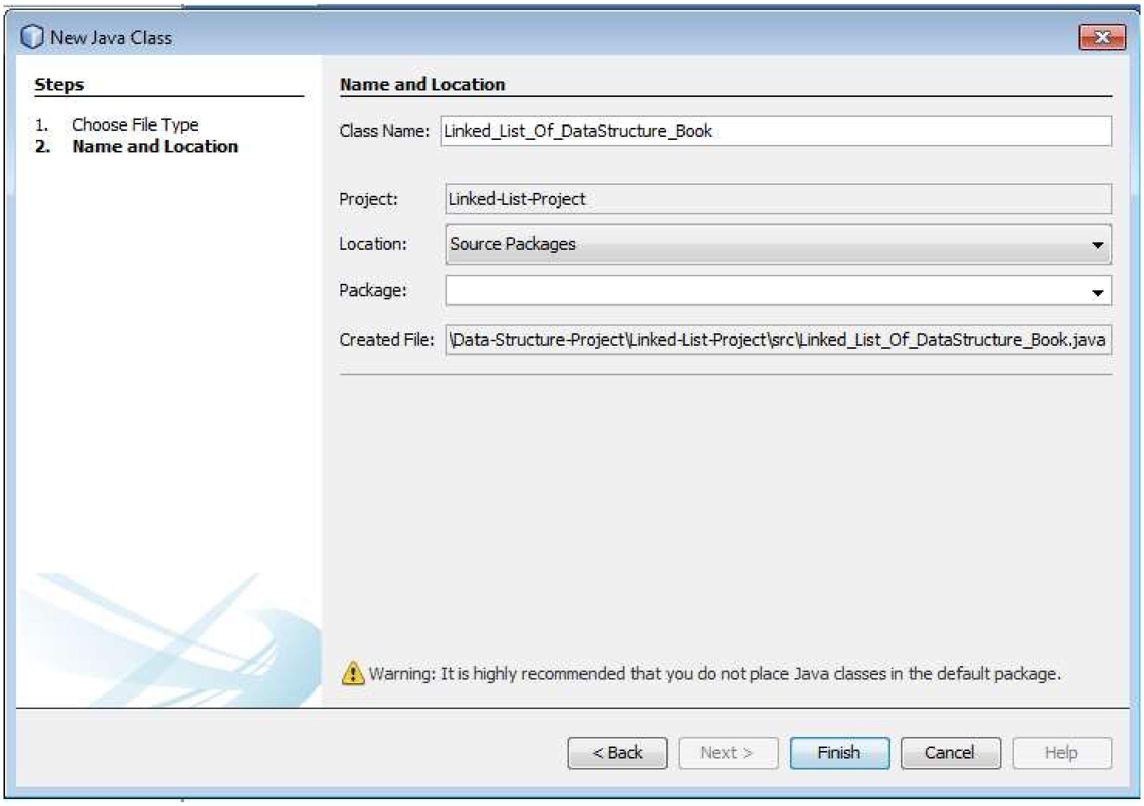
3-سەهمی ماوسەکە ببەرەسەر پۆلی جاڤا Java Main Class و کلیکی لەسەر بکە .



4-ئەم ڕووکارەی خوارەوە دەکرێتەوە:

-لە خانەی ناوی پۆل Class Nameدا، ناوێك بۆ پۆلەکە The Class بنووسە کە دەتەوێ درووستی بکەیت.

-ئەگەر پاکەتێکی تایبەتیمان هەبێت، لەبەشی پاکەت Package دیاری دەکەین، کە لیرەدا، نیمانە و هیچ دیاری ناکەین.



5-کليك لەسەر دووگمەی کۆتاهاتن Finish بکە.

6-بەمەش پۆلێك Class درووست بوو، کە رێگەی سەرەکیشی Main Class تیایە.

```java
/*
 *Data Structure Book By Kurdish Language
 * By: Hemn Mala Karim Barznji
 * Software Development Instructure
 */

/**
 *
 * @author HemnBarznji
 */
public class Linked_List_Of_DataStructure_Book {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

کۆد نووسی بۆ لیستەی پێکەوە بەستراوو

Coding for Linked List

1-سەرەتا پاکەتی جاڤا یوتیلْ Java Util دەهێنینە ناوەوە:

```
import  java.util.*;
```

2-ڕاگەیاندن و بڵاوکردنەوەی لیستەی پێکەوەبەستراوو Linked List Declaration ئەنجام دەدەین، لەناو ڕێگەیی سەرەکی Main Method دا، کە وشەکلیلی LinkedList دەنووسین و، ئەمەش ناوی پۆلی Class لیستەی پێکەوە بەستراوە و، بە دووایدا جۆری زانیاری داغڵکراو دیاری دەکەین کە سترینگە <String> و، پاشان بۆشاییەك دادەنێین، ئینجا ناوێك وەك تەن Object دەنووسین، کە دەبێتە ناوی لیستە پێکەوە بەستراوەکە و، تەنێکە لە پۆلی لیستەی پێکەوە بەستراوو Object of Linked List، و پاشان هێمای = و وشەکلیی نوێ New و، بەهەمان شێوە ناوی پۆلی لیستەی پێکەوە بستراوو و دیاری کردنی جۆری زانیاری داغڵکراوو و جووت کەوانەی کۆتایی و فاریزەیی نوقتەدار لە دووایدا =new LinkedList<String>(); ، بڕوانە کۆدەکەی خوارەوە:

```
LinkedList<String> linkedlist = new LinkedList<String>();
```

3-پاشان بە هۆیی ڕێگەیی زیاد کردنەوە Add Method دانەکان زیاد دەکەین بۆ ناو لیستەی پێکەوەبەستراوویی linkedlist، کە سەرەتا ناوی لیستە پێکەوە بەستراوەکە دەنووسین و پاشان نووقتە دادەنێین و وشەی زیاد کردن add. بە دووایدا دەنووسین و لە نێوان کەوانە و دووانە کۆمای سەرەوەدا (“ ”) ئەو نرخە دەنووسین کە دەمانەوێت داغڵی بکەین:

```
linkedlist.add(“Hemn”);
```

4-بەهەمان شێوە هەموو دانەکانی تر زیاد دەکەین بۆ لیستە پێکەوە بەستراوەکە کە دەمانەوێت:

```
linkedlist.add(“Mala”);
linkedlist.add(“Karim”);
linkedlist.add(“Barznji”);
```

```
linkedlist.add("Software Development Instructor");
```

5-ڕێگەی پیشاندان Display Method دەنووسین، بۆ چاپکردنی لیستەی پێکەوە بەستراوو:

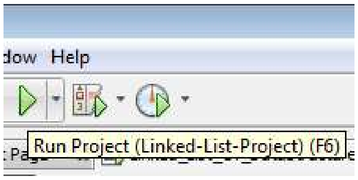
```
System.out.println("Linked List Content: " +linkedlist");
```

ئەمەش هەموو کۆدەکەیە پێکەوە:

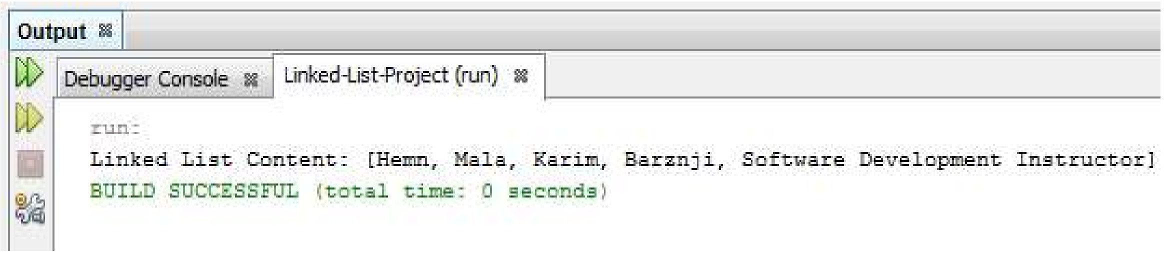
```
/*
 *Data Structure Book – Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
/**
 *
 * @author HemnBarznji
 */
import java.util.*;
public class Linked_List_Of_DataStructure_Book {
   /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {
              /* Linked List Declaration */
           LinkedList<String> linkedlist = new LinkedList<String>();
```

```
        /*add(String Element) is used for adding
      * the elements to the linked list*/
     linkedlist.add("Hemn");
     linkedlist.add("Mala");
     linkedlist.add("Karim");
     linkedlist.add("Barznji");
     linkedlist.add("Software Development Instructor");
      /*Display Linked List Content*/
     System.out.println("Linked List Content: " +linkedlist);
   }
}
```

6-بۆ وەرگێڕان و جێبەجێکردنی Compile & Run ئەو کۆدەی نووسیوومانە کلیك لەسەر دووگمەی جیبەجێکردنی پرۆژە Run Project دەکەین:



7-ئەنجامی کارەکەمان دەبینین:



درووستکەرەکان

Constructor

یەکەم// درووستکەری لیستەی پێکەوە بەستراوو:

ئەم درووستکەرە بەکاردێت بۆ درووستکردنی لیستەیەکی پێکەوە بەستراوویی بەتاڵ Empty.

LinkedList();

دووەم// درووستکەری لیستەی پێکەوە بەستراوو بۆ کۆمەڵە:

ئەم درووستکەرە بەکاردێت بۆ درووستکردنی لیست کە دانەکانی کۆمەڵەیەکی دیاری کراوویی Specified Collection **تیایە.**

LinkedList(Collection<? Extends E> c)

ڕێگەکان

Methods

پۆلی Class **لیستەی پێکەوە بەستراوو** Linked List**، کۆمەلێك ڕێگەی ئامادەکراویی هەیە، کە هەر ڕێگەیەك بەکاردێت بۆ کارێکی دیاری کراوو، بە هۆیانەوە دەتوانین هەموو کردارە پێویستەکان ئەنجامبدەین، بۆیە پێویستە بیانخەینە ڕوو تا بتوانین کردارەکان جێبەجێبکەین و لێیان تێبگەین.**

ڕێگەی زیادکردنی دانەپێکهێنەر لە جۆریی بولییەن

Boolean add(E e);

ئەم ڕێگەیە Method **بەکاردێت بۆ سەربارکردن و زیادکردنی دانەیەکی پێکهێنەر بۆ کۆتایی** End **لیستەی پێکەوە بەستراوویی درووستراوو و ئامادە، کە نرخی ڕاست** True **دەگێڕێتەوە بۆمان.**

نموونە:

```
*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */

/** @author HemnBarznji   */
import java.util.*;
public class Linked_List_Of_DataStructure_Book {
   public static void main(String[] args) {
              /* Linked List Declaration */
            LinkedList<String> linkedlist = new LinkedList<String>();

             /*add(String Element) is used for adding
         * the elements to the linked list*/
        linkedlist.add("Hemn");
        linkedlist.add("Mala");
        linkedlist.add("Karim");
        linkedlist.add("Barznji");
        linkedlist.add("Software Development Instructor");
```

```
 /*Display Linked List Content*/
  System.out.println("Linked List Content: " +linkedlist);
  linkedlist.add("2015-2016");
// print the updated list
 System.out.println("LinkedList:" + linkedlist);
 }
}
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

LinkedList:[Hemn, Mala, Karim, Barznji, Software Development Instructor, 2015-2016]

ڕێگەی زیادکردنی دانەیەکی پێکهێنەر بۆ شوێنێکی دیاری کراو

Void add(int index, E element);

ئەم ڕێگەیە بەکاردێت بۆ زیادکردن و خستنەسەری دانەیەکی پێکهێنەر بۆ لیستەی پێکەوە بەستراوویی ئامادە و، درووستکراو، کە بەهۆی ژمارەیەکی تەواوەوە int، شوێنەکە دیاری دەکەین index تا دانەکەی تیا دابنێین و، پاشان فاریزەیەك دادەنێین و، بە دووای دا دانەکە دەنووسین، واتە ڕێگەکە دوو داغڵکەر Parameter هەیە، و ئەم ڕێگەیە نرخمان بۆ ناگێڕێتەوە.

پێویستە ئەوەمان لەبەرچاو بێت کە لە ژمارە سفرە ((0))، ریزبەندی بۆ شوێنی دانە پێکهێنەرەکانی لیستەی پێکەوەبەستراوو Linked List دەکرێت، بۆیە پێویستە ئێمەش لە ژمارە سفرەوە ((0)) ئەژماری بکەین بۆ دیاری کردنی ئەو شوێنەی دەمانەوێت دانەکەی بخەینە ناو.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */

/** @author HemnBarznji   */
import java.util.*;
public class Linked_List_Of_DataStructure_Book {
   public static void main(String[] args) {
             /* Linked List Declaration */
           LinkedList<String> linkedlist = new LinkedList<String>();

            /*add(String Element) is used for adding
        * the elements to the linked list*/
       linkedlist.add(“Hemn”);
       linkedlist.add(“Mala”);
       linkedlist.add(“Karim”);
       linkedlist.add(“Barznji”);
       linkedlist.add(“Software Development Instructor”);
```

```
 /*Display Linked List Content*/
 System.out.println("Linked List Content: " +linkedlist);

 linkedlist.add(4,"2015-2016");
// print the updated list
  System.out.println("LinkedList:" + linkedlist);
   }
}
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

LinkedList:[Hemn, Mala, Karim, Barznji, 2015-2016, Software Development Instructor]

زیادکردنی کۆمەڵێك دانەی پێکهێنەر بۆ کۆتایی لیستە

Boolean addAll(Collection<?extends E> c)

ئەم ڕێگەیە بەکاردێت بۆ سەربارکردن و زیادکردنی هەموو دانە پێکهێنەرەکانی کۆکراوەیەکی ((کۆمەڵەیەکی)) دیاری کراوو، بۆ کۆتایی لیستە ، بە ڕێکخستن و ڕیزبەندی کۆمەڵە کۆکراوەکە.

داغڵکەری سی C کۆمەڵەیەکە، کە هەموو ئەو دانانەی لە خۆ گرتوە کە زیادکراوە بۆ لیستەکە، ئەم ڕێگەیە نرخی ڕاست True دەگێڕێتەوە، ئەگەر ئەم لیستەیە گۆردرابوو وەکو ئەنجامێك بۆ بانگکردنەوە Call.

ئەگەر کۆمەڵە دیاری کراوەکە بەتاڵبێت ئەوا حاڵەتی هەلاویریی NullPointerException ڕوودەدات.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
/** @author HemnBarznji   */
import java.util.*;
public class Linked_List_Of_DataStructure_Book {
   public static void main(String[] args) {
              /* Linked List Declaration */
            LinkedList<String> linkedlist = new LinkedList<String>();

             /*add(String Element) is used for adding
         * the elements to the linked list*/
        linkedlist.add("Hemn");
        linkedlist.add("Mala");
        linkedlist.add("Karim");
        linkedlist.add("Barznji");
        linkedlist.add("Software Development Instructor");
```

```
        /*Display Linked List Content*/
        System.out.println("Linked List Content: " +linkedlist);

// create a new collection and add some elements
  Collection collection = new ArrayList();
  collection.add("Array");
  collection.add("Linked List");
  collection.add("Stack & Quee");

    // append the collection in the LinkedList
  linkedlist.addAll(collection);
// print the updated list
  System.out.println("LinkedList with Collection:" + linkedlist);
   }
}
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

LinkedList with Collection:[Hemn, Mala, Karim, Barznji, Software Development Instructor, Array, Linked List, Stack & Quee]

زیادکردنی کۆمەڵێك دانەی پێکهێنەر بۆ شوێنێکی دیاری کراوویی لیستە

Boolean addAll(int index, Collection<?extends E> c)

ئەم ڕێگەیە بەکاردێت بۆ سەربارکردن و زیادکردنی هەموو دانە پێکهێنەرەکانی کۆکراوەیەکی ((کۆمەڵەیەکی)) دیاری کراوو، بۆ شوێنێکی دیاری کراوویی لیستەی پێکەوە بەستراوو ، بە ڕێکخستن و ریزبەندی کۆمەڵە کۆکراوەکە، لە ڕێگەی بەکارهێنانی ژمارەیەکی تەواوەوە بۆ دیاری کردنی شوێنەکە.

داغڵکەری سی C کۆمەڵەیەکە، کە هەموو ئەو دانانەی لە خۆ گرتوە کە زیادکراوە بۆ لیستەکە، ئەم ڕێگەیە نرخی ڕاست True دەگێڕێتەوە، ئەگەر ئەم لیستەیە گۆردرابوو وەکو ئەنجامێك بۆ بانگکردنەوە .Call

ئەگەر کۆمەڵە دیاری کراوەکە بەتاڵبێت ئەوا حاڵەتی هەڵاویریی NullPointerException ڕوودەدات. هەروەها ئەگەر شوێنەکە لە دەرەوەی سنووری لیستەی پێکەوە بەستراووبوو، ئەوا IndexOutOfBoundsException ڕوودەدات.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */

/** @author HemnBarznji   */
import java.util.*;
```

```
public class Linked_List_Of_DataStructure_Book {

   public static void main(String[] args) {
            /* Linked List Declaration */
          LinkedList<String> linkedlist = new LinkedList<String>();

           /*add(String Element) is used for adding
        * the elements to the linked list*/
       linkedlist.add("Hemn");
       linkedlist.add("Mala");
       linkedlist.add("Karim");
       linkedlist.add("Barznji");
       linkedlist.add("Software Development Instructor");

        /*Display Linked List Content*/
       System.out.println("Linked List Content: " +linkedlist);

// create a new collection and add some elements
  Collection collection = new ArrayList();
  collection.add("Array");
  collection.add("Linked List");
```

```
collection.add("Stack & Quee");

    // append the collection in the LinkedList
  linkedlist.addAll(4,collection);
// print the updated list
  System.out.println("LinkedList with Collection:" + linkedlist);
   }
}
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

LinkedList with Collection:[Hemn, Mala, Karim, Barznji, Array, Linked List, Stack & Quee, Software Development Instructor]

زیادکردن لە سەرەتای لیستەوە

Void addFirst(E e);

ئەم ڕێگەیە بەکاردەهێنرێت بۆ گەراندنەوەی دانانی دانەیەکی پێکهێنەری دیاری کراوو لە سەرەتایی لیستەکەوە، ئی e ئەو دانە پێکهێنەرەیە کە زیاد دەکرێت.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
```

```
 * Software Development Instructor
 */

/** @author HemnBarznji   */
import java.util.*;
public class Linked_List_Of_DataStructure_Book {

   public static void main(String[] args) {
              /* Linked List Declaration */
           LinkedList<String> linkedlist = new LinkedList<String>();

            /*add(String Element) is used for adding
        * the elements to the linked list*/
       linkedlist.add(“Hemn”);
       linkedlist.add(“Mala”);
       linkedlist.add(“Karim”);
       linkedlist.add(“Barznji”);
       linkedlist.add(“Software Development Instructor”);

        /*Display Linked List Content*/
       System.out.println(“Linked List Content: “ +linkedlist);
```

```
    // add a new element at the beginning of the list
  linkedlist.addFirst("Data Structure Book:");
// print the updated list
  System.out.println("Linked List:" + linkedlist);
   }
}
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Linked List :[Data Structure Book:, Hemn, Mala, Karim, Barznji, Software Development Instructor]

زیادکردن لە کۆتایی لیستەوە

Void addLast(E e);

ئەم ڕێگەیە بەکاردەهێنرێت بۆ گەڕاندنەوەی دانانی دانەیەکی پێکهێنەری دیاری کراوو لە کۆتایی لیستەکەوە، ئی e ئەو دانە پێکهێنەرەیە کە زیاد دەکرێت.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
```

```
/** @author HemnBarznji   */
import java.util.*;
public class Linked_List_Of_DataStructure_Book {

   public static void main(String[] args) {
             /* Linked List Declaration */
           LinkedList<String> linkedlist = new LinkedList<String>();

            /*add(String Element) is used for adding
        * the elements to the linked list*/
       linkedlist.add("Hemn");
       linkedlist.add("Mala");
       linkedlist.add("Karim");
       linkedlist.add("Barznji");
       linkedlist.add("Software Development Instructor");

        /*Display Linked List Content*/
       System.out.println("Linked List Content: " +linkedlist);
  // add a new element at the end of the list
  linkedlist.addLast("2015-2016");
// print the updated list
```

```
  System.out.println("Linked List:" + linkedlist);
   }
}
```

سڕینەوە

Void clear();

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لابردنی Remove هەموو دانەکانی لیستەی پێکەوە بەستراوویی دیاری کراوو. ئەم ڕێگەیە هیچ نرخێك ناگێڕێتەوە.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */

/** @author HemnBarznji   */
import java.util.*;
public class Linked_List_Of_DataStructure_Book {
   public static void main(String[] args) {
             /* Linked List Declaration */
           LinkedList<String> linkedlist = new LinkedList<String>();

            /*add(String Element) is used for adding
```

```
    * the elements to the linked list*/
   linkedlist.add("Hemn");
   linkedlist.add("Mala");
   linkedlist.add("Karim");
   linkedlist.add("Barznji");
   linkedlist.add("Software Development Instructor");

    /*Display Linked List Content*/
   System.out.println("Linked List Content: " +linkedlist);

 // clear the list
  linkedlist.clear();
// print the updated list
  System.out.println("Linked List:" + linkedlist);
   }
}
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Linked List:[]

لەبەرگرتنەوەی لیستەی پێکەوەبەستراوو

Object clone

ئەم رێگەیە بەکاردەهێنرێت بۆ لەبەرگرتنەوەی لیستەی پێکەوەبەستراوویی دیاری کراو، بە گوێرەی پێویست، هیچ داغڵکەرێکی Parameter نییە، و حاڵەتی نائاسایی Exception ڕوونادات.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */


/** @author HemnBarznji   */
import java.util.*;
public class Linked_List_Of_DataStructure_Book {


   public static void main(String[] args) {
             /* Linked List Declaration */
           LinkedList<String> linkedlist = new LinkedList<String>();


            /*add(String Element) is used for adding
        * the elements to the linked list*/
       linkedlist.add(“Hemn”);
```

```
        linkedlist.add("Mala");

        linkedlist.add("Karim");

        linkedlist.add("Barznji");

        linkedlist.add("Software Development Instructor");


         /*Display Linked List Content*/

        System.out.println("Linked List Content: " +linkedlist);


  // create a second LinkedList

   LinkedList list2 = new LinkedList();

   // clone list1

   list2 = (LinkedList) linkedlist.clone();

// print the updated list

   System.out.println("Linked List:" + linkedlist);


    }

}
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Linked List:[Hemn, Mala, Karim, Barznji, Software Development Instructor]

تیابوون

Boolean contain (object O)

ئەم ڕێگەیە نرخی راست True دەگێڕێتەوە، لە حاڵەتی هەبوونی دانەیەکی پێکهێنەر لە لیستەی پێکەوەبەستراوو دا، داغڵکەری ئۆ O ئەو دانە پێکهێنەرەیە کە تاقیدەکرێتەوە و، بۆ ئەوەی بزانین هەیە؟ یان نییە لە لیستەکەدا.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */


/** @author HemnBarznji   */
import java.util.*;
public class Linked_List_Of_DataStructure_Book {


   public static void main(String[] args) {
              /* Linked List Declaration */
           LinkedList<String> linkedlist = new LinkedList<String>();


            /*add(String Element) is used for adding
```

```
 * the elements to the linked list*/
linkedlist.add("Hemn");
linkedlist.add("Mala");
linkedlist.add("Karim");
linkedlist.add("Barznji");
linkedlist.add("Software Development Instructor");

 /*Display Linked List Content*/
System.out.println("Linked List Content: " +linkedlist);

// check if the list contains "H-H"
System.out.println("List contains 'H-H':" + linkedlist.contains("H-H"));

// check if the list contains "Barznji"
System.out.println("List contains 'Barznji':" + linkedlist.contains("Barznji"));
 }
}
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

List contains ‘H-H’:false

List contains ‘Barznji’:true

ڕێگەیی دووبارە کردنەوەی پێچەوانەیی

Iterator<E> descendingIterator()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی دانە پێکهێنەرەکانی لیستەی پێکەوە بەستراوو Linked List لە ڕێکخستنێکی پێچەوانەیی دا، واتە دانەکان لە کۆتاییەوە بۆ سەرەتا ڕێکدەخات.

```
// set Iterator as descending
  Iterator x = linkedlist.descendingIterator();


  // print list with descending order
  while (x.hasNext()) {
  System.out.println(x.next());
  }
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Software Development Instructor

Barznji

Karim

Mala

Hemn

گەراندنەوەی سەرەی لیستەی پێکەوە بەستراوو

E element

ئەم ڕێگەیە بەکاردێت بۆ گێرانەوەی سەرە Head ی لیستەی پێکەوە بەستراوو Linked List، بەلام کرداری لابردن Remove ئەنجام نادات، سەرە Head یەکەم دانەی پێکهێنەریی ناو لیستەی پێکەوە بەستراووە Linked List.

```
// print the head of the list
  System.out.println("Head of list:" + linkedlist.element());
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Head of list:Hemn

گەڕاندنەوەی دانەی پێکهێنەر لە شوێنی دیاری کراوو دا

E get(int index)

ئەم ڕێگەیە بەکاردێت بۆ گەراندنەوەی دانەیەکی دیاری کراوو، کە لە ڕێگەی ژماری دیاری کراوویی شوێنەکەیەوە دەگێردرێتەوە، ئەگەر نا ئاسایی بوون ڕووی داوە و، شوێنەکە لە دەرەوەی سنووری دیاری کراوو بوو ئەوا IndexOutOfBoundsException ڕوودەدات.

// print element at index 3

System.out.println(“Element at index 3 :” + linkedlist.get(3));

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Element at index 3 :Barznji

ڕێگەی گەراندنەوەی یەکەم نرخی لیستەی پێکەوە بەستراوو

E getFirst();

ئەم ڕێگەیە یەکەم نرخی لیستەی پێکەوە بەستراوو دەگێرێتەوە، هیچ داغڵکەرێکی Parameter نییە، بەڵام ئەگەر لیستەکە بەتاڵ بوو، ئەوا NoSuchElementException ڕوودەدات.

// print the first element of the list

System.out.println(“First Element :” + linkedlist.getFirst());

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

First Element :Hemn

ڕێگەی گەراندنەوەی کۆتا دانەی لیستەی پێکەوە بەستراوو

E getLast()

ئەم ڕێگەیە کۆتا نرخی لیستەی پێکەوە بەستراوو دەگێرێتەوە، هیچ داغڵکەرێکی Parameter نییە، بەڵام ئەگەر لیستەکە بەتاڵ بوو، ئەوا NoSuchElementException ڕوودەدات.

// print the last element of the list

System.out.println("Last Element :" + linkedlist.getLast());

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Last Element :Software Development Instructor

ڕێگەی دیاری کردنی شوێنی دانەی پێکهێنەر

Int indexOf(Object O)

ئەم ڕێگەیە بەکاردێت بۆ گەراندنەوەی شوێنی دانەی پێکهێنەر Index، کە یەکەم جار هاتووە لەناو لیستەی پێکەوە بەستراوودا، چونکە لە وانەیە ئەو دانەیە چەند جارێك نووسرابێت و هەبێت، بەلام ئەگەر دانەکە نەبوو ئەوا نرخی ‎-1‎ بۆ شوێنەکەی دەگێرێتەوە کەمانای نەبوونی ئەو دانەیەیە لە لیستەکەدا.

// get the index for "Barznji"

System.out.println("Index for Chocolate:" + linkedlist.indexOf("Barznji"));

// get the index for "Siwayly"

System.out.println("Index for Coffee:" + linkedlist.indexOf("Siwayly"));

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Index for Chocolate:3

Index for Coffee:-1

ڕێگەی دیاری کردنی شوێنی دووایین دانەی پێکهێنەر

Int lastIndexOf(Object O)

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی شوێنی دانەی پێکهێنەر Index، کە دووایین جار هاتووە لەناو لیستەی پێکەوە بەستراوودا، چونکە لە وانەیە ئەو دانەیە چەند جارێك نووسرابێت و هەبێت، بەڵام ئەگەر دانەکە نەبوو ئەوا نرخی -1 بۆ شوێنەکەی دەگێڕێتەوە کەمانای نەبوونی ئەو دانەیەیە لە لیستەکەدا.

```
// get the last index for "Hemn"
System.out.println("Index for Hello:" +
linkedlist.lastIndexOf("Hemn"));


// get the index for "Siwaily"
System.out.println("Index for Coffee:" +
linkedlist.indexOf("Siwaily"));
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor, Hemn]

Index for Hello:5

Index for Coffee:-1

ڕێگەی دووبارە کردنەوەی دانەکان لە شوێنی دیاری کراوەوە

ListIterator<E> listIterator(int index)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ گەڕاندنەوەی لیستێکی دووبارە کراوە List Iterator لە شوێنی دیاری کراوەوە Index، واتە لیستە دووبارە کراوەکە لە شوێنی دیاری کراوەوە Index دەستپێدەکات. بەڵام ئەگەر شوێنەکە لە دەرەوەی سنوور و ماوەی دیاری کراوو بوو ئەوا IndexOutOfBoundsException ڕوودەدات.

```
// set Iterator at specified index
  Iterator x = linkedlist.listIterator(3);


  // print list with the iterator
  while (x.hasNext()) {
  System.out.println(x.next());
  }
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor, 2015-2016]

Barznji

Software Development Instructor

2015-2016

زیاد کردنی دانەیەك وەك کلکی لیستەی پێکەوە بەستراوو

Boolean offer (E e)

ئەم رێگەیە بەکاردەهێنرێت بۆ زیاد کردنی دانەیەکی دیاری کراوو، وەکو دانەی کۆتایی Last Element لیستەی پێکەوە بەستراوو.

```
// offer a new element
  linkedlist.offer("2015-2016");
  // print the new list
  System.out.println("Linked List:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Linked List:[Hemn, Mala, Karim, Barznji, Software Development Instructor, 2015-2016]

رێگەی زیاد کردنی دانەیەك لەبەردەم لیستەی پێکەوە بەستراوو

Boolean offerFirst(E e)

ئەم رێگەیە بەکاردێت بۆ زیادکردنی دانەیەکی پێکهێنەر Element لەبەردەم لیستەی پێکەوە بەستراوو Linked List ی دیار ی کراوودا.

```
 // offer a new element as the head of the list
  linkedlist.offerFirst("Data Structure");
```

```
// print the new list
System.out.println("LinkedList:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

LinkedList:[Data Structure, Hemn, Mala, Karim, Barznji, Software Development Instructor]

ڕێگەی زیاد کردنی دانەیەك لە کۆتایی لیستەی پێکەوە بەستراو

Boolean offerLast(E e)

ئەم ڕێگەیە بەکاردێت بۆ زیادکردنی دانەیەکی پێکهێنەر Element لە کۆتایی لیستەی پێکەوە بەستراوو Linked List ی دیار ی کراوودا.

```
// offer a new element as the tail of the list
linkedlist.offerLast("2015-2016");
// print the new list
System.out.println("LinkedList:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Linked List:[Hemn, Mala, Karim, Barznji, Software Development Instructor, 2015-2016]

ڕێگەیی گەڕاندنەوەی نرخی سەرە

E Peek ()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی نرخی دانەی سەرە Head ، لیستەی پێکەوە بەستراوو ، یان گەڕاندنەوەی نرخی بەتال Null ئەگەر لیستەکە بەتاڵ Empty بوو.

```
// peek at the head of the list
System.out.println("Head of the list:" + linkedlist.peek());
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Head of the list:Hemn

ڕێگەیی گەڕاندنەوەی نرخی یەکەم دانە

E PeekFirst ()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی نرخی یەکەم دانەی ، لیستەی پێکەوە بەستراوو ، یان گەڕاندنەوەی نرخی بەتال Null ئەگەر لیستەکە بەتاڵ Empty بوو.

```
// peek at the first element
System.out.println("First element of the list:" +
linkedlist.peekFirst());
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

First element of the list:Hemn

ڕێگەیی گەراندنەوەی نرخی کۆتا دانە

E PeekLast ()

ئەم ڕێگەیە بەکاردێت بۆ گەراندنەوەی نرخی کۆتا دانەی ، لیستەی پێکەوە بەستراوو ، یان گەراندنەوەی نرخی بەتال Null ئەگەر لیستەکە بەتاڵ Empty بوو.

```
  // peek at the last element
  System.out.println("Last element of the list:" +
linkedlist.peekLast());
```

ئەنجلم//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Last element of the list:Software Development Instructor

ڕێگەی گەراندنەوە و لابردنی سەرە

E Poll()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوە و لابردنی سەرە ((یەکەم دانەی پێکهێنەریی)) لیستەی پێکەوە بەستراوو.

```
// retrieve and remove the head of the list
  System.out.println("Head element of the list:" + linkedlist.poll());


  // print the list
```

```
  System.out.println("Linked List:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Head element of the list:Hemn

Linked List:[Mala, Karim, Barznji, Software Development Instructor]

رێگەی گەڕاندنەوە و لابردنی یەکەم دانە

E PollFirst()

ئەم رێگەیە بەکاردێت بۆ گەڕاندنەوە و لابردنی یەکەم دانەی پێکهێنەریی لیستەی پێکەوە بەستراوو. یان گەڕاندنەوەی نرخی هیچ شتێك Null ئەگەر لیستەکە بەتاڵ Empty بوو.

```
// retrieve and remove the head of the list
  System.out.println("First element of the list:" + linkedlist.pollFirst());


  // print the list
  System.out.println("Linked List:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

First element of the list:Hemn

Linked List:[Mala, Karim, Barznji, Software Development Instructor]

ڕێگەی گەڕاندنەوە و لابردنی کۆتا دانە

E pollLast()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوە و لابردنی کۆتا دانەی پێکهێنەریی لیستەی پێکەوە بەستراوو. یان گەڕاندنەوەی نرخی هیچ شتێك Null ئەگەر لیستەکە بەتاڵ Empty بوو.

```
// retrieve and remove the Last  of the list
  System.out.println("Last element of the list:" + linkedlist.pollLast());


  // print the list
  System.out.println("Linked List:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Last element of the list:Software Development Instructor

Linked List:[Hemn, Mala, Karim, Barznji]

ڕێگەی دەرهێنانەوەی دانەی پێکهێنەر

E pop()

ئەم ڕێگەیە بەکاردەهێنرێت بۆ دەرهێنانەوەی Pop دانەی پێکهێنەر Element لە ریزەخانەی ستوونی ((ریزە ڕەفە)) Stack دا، کە بە لیستەی پێکەوە بەستراوو Linked List درووستکرابێت.

```
// pop the list
  System.out.println("Pop element in the list:" + linkedlist.pop());
```

```
// print the list
System.out.println("LinkedList:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Pop element in the list:Hemn

LinkedList:[Mala, Karim, Barznji, Software Development Instructor]

ڕێگەی تیادانی دانەی پێکهێنەر

Void push(E e)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ کرداریی دانان و خستنە ناوی push دانەی پێکهێنەر بۆ ڕیزەخانە ((ریزە ڕەفە)) ی دانانی زانیاریی Stack کە بە لیستەی پێکەوە بەستراوو دروستکرابێت، هەروەها هیچ جۆرە نرخێك ناگێرێتەوە.

```
// push Element the list
linkedlist.push("2015-2016");


// print the list
System.out.println("LinkedList:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

LinkedList:[2015-2016, Hemn, Mala, Karim, Barznji, Software Development Instructor]

ڕێگەی گەڕاندنەوە و لابردنی سەرە

E Remove()

ئەم ڕێگەیە بەکاردەهێنرێت بۆ گەڕاندنەوە Retrieve و لابردنی Remove سەرە Head ((یەکەم دانە ی پێکهێنەر First Element)) ی لیستەی پێکەوە بەستراوو Linked List.

```
// remove the head of the list
linkedlist.remove();


// print the list
System.out.println("LinkedList:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

LinkedList:[Mala, Karim, Barznji, Software Development Instructor]

ڕێگەیی لابردنی دانەیەکی پێکهێنەری دیاری کراوو بە هۆی شوێنەکەیەوە

E remove(int index)

ئەم ڕێگەیە بەکاردێت بۆ لابردنی دانەیەکی پێکهێنەر لە لیستەی پێکەوە بەستراوو Linked List بە گوێرەیی دیاری کردنی شوێنەکەی، کە داغڵکەرەکەی Parameter بریتییە لە شوێنی دانەکە Index.

```
// remove the element at index 3
```

```
System.out.println("Element to be removed:" +
linkedlist.remove(3));


// print the list
System.out.println("LinkedList:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Element to be removed:Barznji

LinkedList:[Hemn, Mala, Karim, Software Development Instructor]

ڕێگەی لابردنی دانەیەکی پێکهێنەر لە یەکەم دەرکەوتن دا

Boolean remove(Object o)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لابرنی دانەیەکی پێکهێنەر لەیەکەم دەرکەوتنی دا، چونکە لە وانەیە هەمان دانە چەند جارێکی تر لە لیستەکەدا هاتبێت و، لە شوێنی جیاوازدا، دەرکەوتبێت، ئەگەر دانەکە هەبوو لای دەبات و نرخی ڕاست True دەگێڕێتەوە.

```
    // remove "Barznji"
System.out.println("Barznji is in the list:" +
linkedlist.remove("Barznji"));


// print the list
System.out.println("LinkedList:" + linkedlist);
```

ڕێگەیی لابردنی یەکەم دانەی لیستەی پێکەوە بەستراوو

E removeFirst()

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لابردن و گەڕاندنەوەی یەکەم دانەی پێکهێنەر لە لیستەی پێکەوە بەستراوو Linked Listدا. ئەگەر لیستەکە بەتاڵ بوو ئەوا NoSuchElementException ڕوودەدات.

```
// remove the first element
System.out.println("First element:" + linkedlist.removeFirst());


// print the list
System.out.println("Linked List:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor, Barznji]

Element found:true

LinkedList:[Hemn, Mala, Karim, Software Development Instructor, Barznji]

ڕێگەی لابردنی دانەیەکی پێکهێنەر لە یەکەم دەرکەوتن دا

Boolean removeFirstOccurrence(Object o)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لابرنی دانەیەکی پێکهێنەر لەیەکەم دەرکەوتنی دا، چونکە لە وانەیە هەمان دانە چەند جارێکی تر لە لیستەکەدا هاتبێت و، لە شوێنی جیاوازدا، دەرکەوتبێت، ئەگەر دانەکە هەبوو لای دەبات و نرخی ڕاست True دەگێڕێتەوە. بەڵام کاتێك لیستەی پێکەوە بەستراوو Linked

List پێچەوانە کرابێتەوە Traversing لە سەرەوە Head بۆ کلك Tail. ئەگەر دانەکە لە لیستەکەدا نەبوو، لیستەکە ناگۆڕێت.

```
 // remove the first "Barznji"
boolean found = linkedlist.removeFirstOccurrence("Barznji");
System.out.println("Element found:" + found);


// print the list
System.out.println("LinkedList:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Element found:true

LinkedList:[Hemn, Mala, Karim, Software Development Instructor]

ڕێگەیی لابردنی کۆتا دانەی لیستەی پێکەوە بەستراوو

E removeLast()

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لابردن و گەراندنەوەی کۆتا دانەی پێکهێنەر لە لیستەی پێکەوە بەستراوو Linked Listدا. ئەگەر لیستەکە بەتال بوو ئەوا NoSuchElementException ڕوودەدات.

```
// remove the last element
System.out.println("Last element:" + linkedlist.removeLast());
```

```
// print the list
System.out.println("LinkedList:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor, Barznji]

Last element:Barznji

LinkedList:[Hemn, Mala, Karim, Barznji, Software Development Instructor]

ڕێگەی لابردنی دانەیەکی پێکهێنەر لە دواین دەرکەوتن دا

Boolean removeLastOccurrence(Object o)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لابرنی دانەیەکی پێکهێنەر لە دووایین دەرکەوتنی دا، چونکە لە وانەیە هەمان دانە چەند جارێکی تر لە لیستەکەدا هاتبێت و، لە شوێنی جیاوازدا، دەرکەوتبێت، ئەگەر دانەکە هەبوو لای دەبات و نرخی ڕاست True دەگێڕێتەوە. بەڵام کاتێك لیستەی پێکەوە بەستراوو Linked List پێچەوانە کرابێتەوە Traversing لە سەرەوە Head بۆ کلك Tail. ئەگەر دانەکە لە لیستەکەدا نەبوو، لیستەکە ناگۆڕێت.

```
// remove the last "Barznji"
boolean found = linkedlist.removeLastOccurrence("Barznji");
System.out.println("Element found:" + found);


// print the list
System.out.println("LinkedList:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor, Barznji]

Element found:true

LinkedList:[Hemn, Mala, Karim, Barznji, Software Development Instructor]

ڕێگەیی لە جیاتی دانان

E set(int index, E element)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لە جیاتی دانانی دانەیەك لە شوێنێکی دیاری کراوو دا، لە لیستەی پێکەوە بەستراوو Linked Listدا، لەگەڵ دانەیەکی دیاری کراووی تر دا شوێنگۆرکێکە ئەنجام دەدرێت.

ئەم ڕێگەیە دوو داغڵکەری Parameter هەیە و، داغڵکەری index شوێنی ئەو دانە پێکهێنەرەیە کە دەمانەوێت لای بەرین و، داغڵکەری Element ئەو دانەیەیە کە دەمانەوێ بیخەینە شوێن دانە شوێن دیاری کراوەکە.

```
// set "Barziny" at index 3
System.out.println("Object to be replaced:" + linkedlist.set(3, "Barziny"));


// print the list
System.out.println("LinkedList:" + linkedlist);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor, Barznji]

Object to be replaced:Barznji

LinkedList:[Hemn, Mala, Karim, Barziny, Software Development Instructor, Barznji]

ڕێگەیی گەڕاندنەوەی نرخی ژمارەیی دانەکانی لیستەی پێکەوە بەستراوو

Int size()

ئەم ڕێگەیە بەکاردێت بۆ زانینی حەجمی لیستەی پێکەوە بەستراوو Linked List کە دەکاتە ژمارەی دانە پێکهێنەرەکانی ناو لیستەکە.

```
// print the size of the list
System.out.println("List size:" + linkedlist.size());
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor, Barznji]

List size:6

لە بەرگرتنەوەی لیستەی پێکەوە بەستراوو بۆ ناو Array

Object[] toArray()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی ئەرەیی Array کە هەموو دانەکانی لیستەی پێکەوە بەستراوویی تیابێت بە ڕێکخستنی لە یەکەمەوە First بۆ کۆتایی Last، ئەم ڕێگەیە کاردەکات وەك پردێك لە نێوا ئەی پی ئای APIs ئەرەی بنیچینەیی Array – Based و کۆمەڵەی بنچینەیی Collection – Based

```
// create an array and copy the list to it
  Object[] array = linkedlist.toArray();

  // print the array
  for (int i = 0; i < linkedlist.size(); i++)
{
  System.out.println("Array:" + array[i]);
}
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor, Barznji]

Array:Hemn

Array:Mala

Array:Karim

Array:Barznji

Array:Software Development Instructor

Array:Barznji

له بەرگرتنەوەی لیستەی پێکەوە بەستراوو بۆ ناو Array

<T> T[] toArray(T[]a)

ئەم رێگەیە بەکاردێت بۆ گەراندنەوەی ئەرەیی Array کە هەموو دانەکانی لیستەی پێکەوە بەستراوویی تیابێت بە رێکخستنی لە یەکەمەوە First بۆ کۆتایی Last، بەڵام حەجم دیاری دەکەین و، بەمەش کاتی لەبەر گرتنەوەکە یەکسانە بە ژمارەی دانەکان و دیاری کراوە و ورد و شیاوترە.

```
// create an array and copy the list to it
  Object[] array = linkedlist.toArray(new Object[4]);


  // print the array
  for (int i = 0; i < linkedlist.size(); i++) {
  System.out.println("Array:" + array[i]);
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor, Barznji]

Array:Hemn

Array:Mala

Array:Karim

Array:Barznji

Array:Software Development Instructor

Array:Barznji

سوڕەکانی لیستەی پێکەوە بەستراوو

Loops Linked List

دەتوانین هەموو جۆرەکانی سوورەکان لەگەڵ لیستەی پێکەوە بەستراوو Linked List بەکاربهێنین، بۆیە لەم نموونەیەدا، چاپ کردن Print ئەنجام دەدەین، بە بەکارهێنانی سورەکان Loops.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */


/** @author HemnBarznji   */
import java.util.*;
public class Linked_List_Of_DataStructure_Book {
   public static void main(String[] args) {
             /* Linked List Declaration */
           LinkedList<String> linkedlist = new LinkedList<String>();


            /*add(String Element) is used for adding
        * the elements to the linked list*/
       linkedlist.add("Hemn");
       linkedlist.add("Mala");
```

```
  linkedlist.add("Karim");

  linkedlist.add("Barznji");

  linkedlist.add("Software Development Instructor");

  linkedlist.add("Barznji");


   /*Display Linked List Content*/

  System.out.println("Linked List Content: " +linkedlist);




 /*for loop*/

System.out.println("**For loop**");

for(int num=0; num<linkedlist.size(); num++)

{

    System.out.println(linkedlist.get(num));

}


/*Advanced for loop*/

System.out.println("**Advanced For loop**");

for(String str: linkedlist)

{

    System.out.println(str);
```

```
    }

    /*Using Iterator*/
    System.out.println("**Iterator**");
    Iterator i = linkedlist.iterator();
    while (i.hasNext()) {
        System.out.println(i.next());
    }

    /* Using While Loop*/
    System.out.println("**While Loop**");
    int num = 0;
    while (linkedlist.size() > num) {
        System.out.println(linkedlist.get(num));
        num++;
    }

  }
}
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor, Barznji]

For loop

Hemn

Mala

Karim

Barznji

Software Development Instructor

Barznji

Advanced For loop

Hemn

Mala

Karim

Barznji

Software Development Instructor

Barznji

Iterator

Hemn

Mala

Karim

Barznji

Software Development Instructor

Barznji

While Loop

Hemn

Mala

Karim

Barznji

Software Development Instructor

Barznji

جیاوازی نێوان

لیستەی ریزکراوە و لیستەی پێکەوە بەستراوو

Difference between Array List and Linked List

لیستەی ئەرەیی Array List	لیستەی پێکەوە بەستراوو Linked List
لە ناوەرۆکدا، ئەرەیی داینەمیکی Dynamic Array بەکاردەهێنێت بۆ پاشەکەوت کردنی دانە پێکهێنەرەکان.	لە ناوەرۆك دا، لیستەی پێکەوە بەستراوویی دوو ئاراستە Doubly Linked List بەکار دەهێنێت بۆ پاشەکەوت کردنی دانە پێکهێنەرەکان.
جێبەجێکردن لەگەڵ لیستەی ئەرەیی، خاوە Slow چونکە لە بەرئەوەی لە ناوەرۆکدا، ئەرەیی Array بەکاردەهێنێت، ئەگەر هەر دانەیەك لابرێت Remove لە ئەرەیەکەدا، هەموو بتەکانی میمۆری شیفت Shift دەکەن.	جێبەجێکردن لەگەڵ لیستەی پێکەوە بەستراوو، خێراترە Faster، لە لیستەی ئەرەیی، لەبەرئەوەیی لیستەی پێکەوە بەستراوویی دوو ئاراستە بەکار دەهێنێت و، شێفتکردنی بتەکان Bit Shifting داواکراوو و پێویست نی یە ڕووبات لە میمۆری دا.
پۆلی ئەرەیی Class of Array کاردەکات وەکو لیستە List، چونکە تەنها لیستە List جێبەجێدەکات Implement.	پۆلی لیستەی پێکەوە بەستراوو Class of Linked List، کاردەکات وەکو لیستە List و کیوو Queue، لەبەر ئەوەی ڕووکاری لیست List و Deque جێبەجێدەکات.
لیستەی ئەرەیی Array List باشترە بۆ پاشەکەوت کردن و دەستپێگەیشتنی زانیاریی Sorting and Accessing Data	لیستەی پێکەوە بەستراوو Linked List باشترە بۆ چارەسەر کردنی زانیاریی Better for Manipulating Data

ئەمانە جیاوازییە گشتی و گرنگەکانن و، دەتوانین بە ورد تر لیستەی ئەرەیی و لیستەی پێکەوە بەستراوو بەراوورد بکەین، لە چەند ڕوویەکەوە، وەکو:

- گەڕان Search: کرداریی گەران لە لیستەی ئەرەیی Array Listدا، خێراترە بە بەراورد بە کرداریی گەران Search Operation لە لیستەی پێکەوە بەستراوو Linked List، چوستی و جیبەجێکردن Performance ی ئەرەی لیست Array List بریتییە لە O(1) ،

بەڵام چوستی و جێبەجێکردن Performance ی لیستەی پێکەوە بەستراوو Linked List بریتییە لە O(n)،

- سرینەوە Deletion: چوستی و جێبەجێکردنی Performance، کرداری لابردن Remove لە لیستەی پێکەوە بەستراوو Linked List دا، پێویستی بە O(1) هەیە، بەڵام هەمان کردار لیستەی ئەرەیی Array List پێویستی بە O(n) هەیە لە خراپترین حاڵەتدا، و O(1) لە باشترین حاڵەت دا.
- دانان ((خستنە ناو)) Insert: چوستی و جێبەجێکردنی Performance لیستەی پێکەوە بەستراوو Linked List بۆ ئەم کردارە، بریتییە لە O(1)، بەڵام بۆ ئەرەی لیست بریتییە لە O(n) لە خراپترین حاڵەت دا.
- سەرباربوونی میمۆری Memory Overhead: ئەرەی لیست Array List زانیاریی و شوێنی Index هەیە، بەڵام لیستەی پێکەوە بەستراوو Linked List، زانیاری Data و ئاماژە پێدەرەکانی Pointers هەیە. بۆ گرێکانی تەنیشت Neighbor Node، بۆیە بەکارهێنانی میمۆری Memory Consumption زیاترە لە لیستەی پێکەوە بەستراوو Linked Listدا.

بڕووانە نموونەکە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */


/** @author HemnBarznji   */
import java.util.*;
```

```
public class Linked_List_Of_DataStructure_Book {

   public static void main(String[] args) {
              /* Linked List Declaration */
           LinkedList<String> linkedlist = new LinkedList<String>();

            /*add(String Element) is used for adding
        * the elements to the linked list*/
       linkedlist.add("Hemn");
       linkedlist.add("Mala");
       linkedlist.add("Karim");
       linkedlist.add("Barznji");
       linkedlist.add("Software Development Instructor");
        /*Display Linked List Content*/
       System.out.println("Linked List Content: " +linkedlist);


     ArrayList<String> arrayList = new ArrayList<String>();//creating
//Array List
  arrayList.add("HS");//adding object in Array List
  arrayList.add("Company");
 arrayList.add("For IT");
```

```
  arrayList.add("Solutions");

  System.out.println("Array List: "+arrayList);

    }
}
```

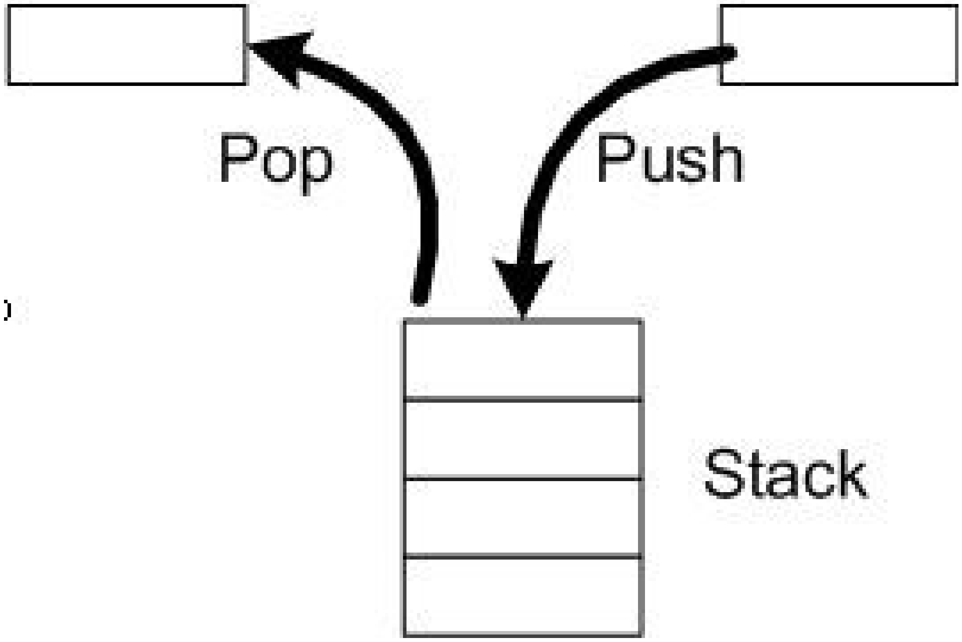
ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Array List: [HS, Company, For IT, Solutions]

شارای ڕەفەیی

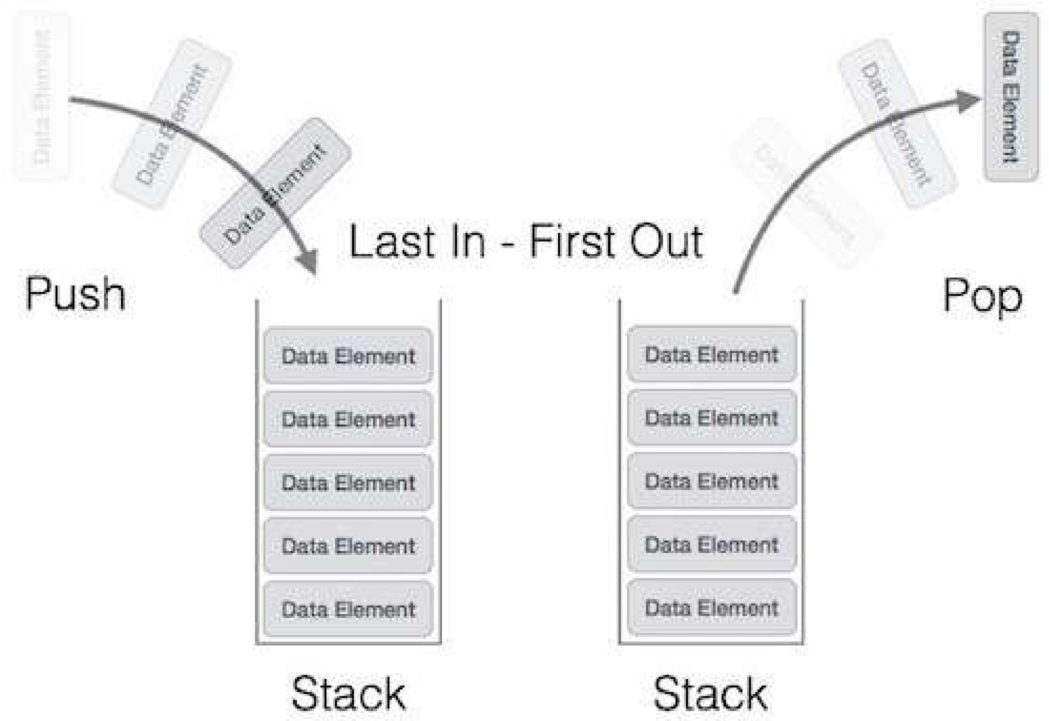
Stack



بیرۆکەی بنچینەیی شارای ڕەفەیی

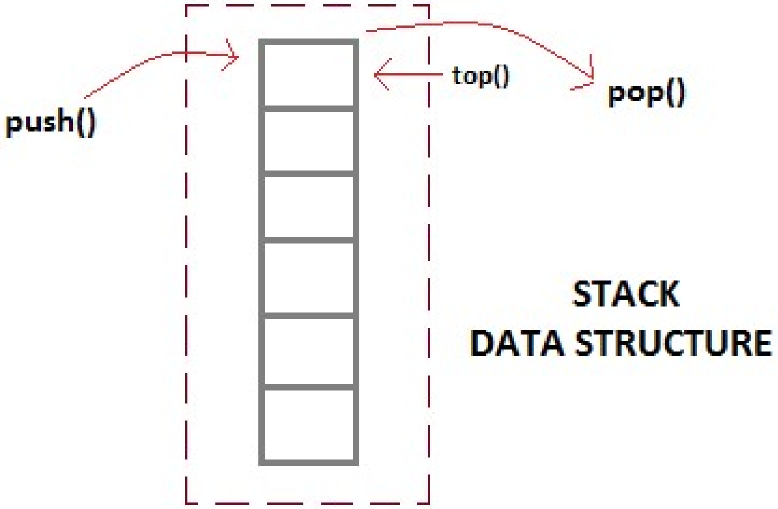
Basic Concept of Stack

شارایی ڕەفەیی Stack بەکاردێت بۆ پاشەکەوت کردنی زانیاری و شێوەی پێکهاتنی بە جۆرێکە، کە یەکەم دانەی داغڵبوو First In، دووواین جار دێتەوە دەر Last Out، یاخود دەتوانین بلێین: ئەوەی دووا جار دێت Last In، یەکەمجار دەرواتە دەرەوە First Out:



شارایی ڕەفەیی Stack جۆری زانیاری پوختەیە ((ڕووتە)) Abstract Data Type، کە سەرەتا تواناکەی دیاری دەکرێت، جۆرێکی سادەیی شێوەی پێکهاتنی زانیارییە و، کە ڕێگە بە زیادکردنی Adding دانەکان و، سرینەوە و لابردنەوەی Remove دانەکان دەدات، بە ڕێکخستنێکی دیاری کراوو.

هەرکاتێك دانە زیاد دەکەین، دەچێتە بەشی سەرەوەی Top شارای ڕەفەیی Stack، هەروەها تەنها ئەو دانەیەی دەکەوێتە سەرەوە دتوانرێت لابرێت و، دانەکانی تریش بە دووایدا. ئەمەش ڕێکخستنێکی تایبەتی یە و، ڕێسایەکی تایبەتی هەیە کە بە لیفۆ LIFO ناسراوە.



نموونەیەکی ڕوون و راستەقینە وەك رەفەیی کتیبخانە، کە کتیبەکانی تێدا دادەنرێت، وەك لە وێنەکەدا، دیارە:

ئەم کتیبانە چۆن ریکخراوون لە رەفەدا:

- کتیبەکان هەڵگیراون پارێزراوون یەك لەسەر یەك.
- ئەو کتیبەی یەکەم جار دانراوە، دوواجار دەردەهێنرێتەوە.
- ئەو کتیبەی دووا جار دانراوە، یەکەم جار بەکاردەهێنرێت و دەردەهێنرێتەوە.

نموونەی باو

Common Example

لە ماڵی هەر یەکێمان دا، چەنج کورسییەك هەیە و، کاتێك یەك بە یەك دایان دەنێین لەسەر یەکتری و، یەکەم دانە کە کەوتۆتە ژێرەوە و یەکەم جار دامان ناوە، و ئەوانی ترمان خستۆتە سەر، دوواتر دەبێت دووایین جار بەکاری بهێنین .

بە پێچەوانەشەوە ئەوەی دووایین جار دامانناوە لەسەرەوەی هەموو کورسییەکان، دەتوانین یەکەم جار هەڵیبگرینەوە و بەکاری بهێنینەوە، یان بلێین لای ببەینەوە لەسەر ریزە کورسییەکەی تر.

رووخسارە بنچینەییەکانی شارای ڕەفەیی

Basic Features of Stack

1- شارای ڕەفەیی Stack لیستەیەکی رێکخراوە Ordered List لە جۆرێکی لێچووی زانیاری Same Data Type واتە هەموو زانیارییەکان یەك جۆربن، بۆ نموونە هەمووی ژمارە بێت، یان هەمووی سترینگ String بێت، یان ..... .

2- شارای ڕەفەیی Stack شێوەی پێکهاتنی لیفۆیی LIFO هەیە، واتە ئەوەی دووا جار دێتە ناوەوە Last In، یەکەم جار دەچێتە دەرەوە First Out.

3- کرداری خستنە ناو Push Function بەکاردێت بۆ دانانی Insert دانەیەکی نوێ بۆ ناو شارای ڕەفەیی Stack و، کرداری دەرهێنانەوەش Pop Function بەکاردێت بۆ سرینەوەی Delete دانەکانی ناو شارای ڕەفەیی Stack، هەردوو کرداری خستنە ناو و سرینەوە Insertion and Deletion لەسەر یەکەم دانەی کۆتایی ئەنجام دەدرێت کە پێی دەڵێن سەرە Top.

4- ئەگەر شوێنی بەتاڵ هەبێت ئەوا دانەی نوێ زیاد دەبێت لەکاتی زیاد کردن دا، بە پێچەوانەوە سەر دەکەوێت و جێگەی نابێتەوە Over Flow.

جێبەجێکردنەکانی شارای ڕەفەیی

Applications of Stack

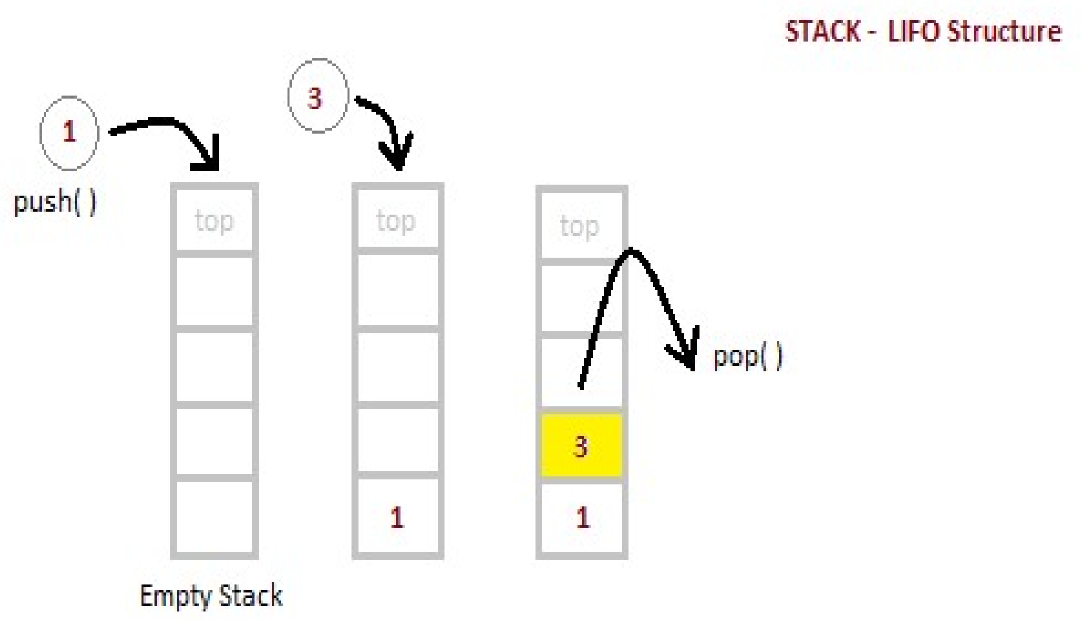
سادەترین جێبەجێکردنی شارای ڕەفەیی Stack پێچەوانە کردنەوەی وشەیە، بە جۆرێك ئێمە وشەکە داغڵ دەکەین، بە پاڵ پێوەنانی پیت بە پیتی وشەکە، پاشان دەرهێنانەوەی پیت بە پیتی وشەکە و بەمەش وشەکە پێچەوانە دەبێتەوە.

بەکارهێنانێکی تری شارای ڕەفەیی Stack: دابەشکاری Parsing، گۆرینی برە و گوزارشتەکان Expression Conversion، هەروەها بەکارهێنانی تریش.

دروستکردنی شارای ڕەفەیی

Stack Implementation

شارای ڕەفەیی Stack بە سادەیی دروستدەکرێت بە بەکارهێنانی ئەرەیی Array یان لیستەی پێکەوەبەستراوو Linked List، بەڵام حەجمەکەی سنوور دارە، واتە دەبێت حەجمەکەی جێگیربێت Fixed.



کردارە بنچینەییەکان

Basic Operations

کردارە بنچینەییەکانی شارای ڕەفەیی ئەمانەی لای خوارەوەن، کە بە نموونەوە ڕوونیان دەکەینەوە:

کرداری خستنە ناو Push() Operation

خستنە ناو ((پاڵنان بۆ ناو)) Push (): ئەم کردارە بەکاردێت بۆ زانیاری خستنە ناو شارای ڕەفەیی Stack و، پاشەکەوت کردنی Store زانیاریی. واتە دانەکان دەخاتە ناو شارای ڕەفەیی.

بڕوانە ئەلگۆریسمەکە:

```
begin procedure push: stack, data
  if stack is full
    return null
  endif
  top ← top + 1
  stack[top] ← data
end procedure
```

هەنگاوەکان و کردارەکان بۆ داغڵکردن و خستنە ناوی دانەیەك بۆ ناو شارای ڕەفەیی Stack ئەمانەی لای خوارەوەن:

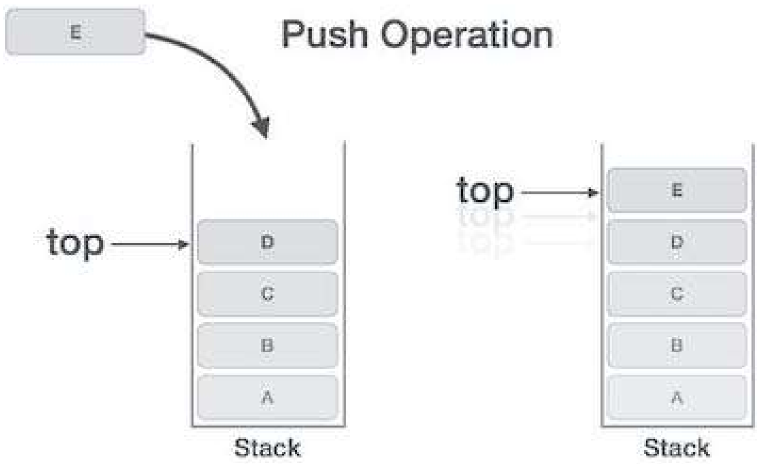
هەنگاوی 1: پشکنینی شارای ڕەفەیی ئەگەر پڕبێت Check If Stack is Full.

هەنگاوی 2: ئەگەر شارای ڕەفەیی پڕبێت، هەڵە Error ڕوودەدات و دەرچوون Exit دەبێت.

هەنگاوی3:ئەگەر شارای ڕەفەیی Stack پڕ نەبێت، دانەی سەرەوە Top زیاد دەکات و بۆشایی بۆ دانەی نوێ دەبێت.

هەنگاوی4: دانەی زانیاری زیاد دەبێت بۆ ناو شارای ڕەفەیی Stack، و دەبێتە دانەی سەرەوە Top.

هەنگاوی5: سەرکەوتووانە کارەکە جیبەجێدەبێت.



کرداری دەرهێنانەوە ((لابردن)) POP() Operation

دەرهێنانەوە Pop(): ئەم کردارە بەکاردێت بۆ لابردنەوەی Remove دانەکانی ناو شارای ڕەفەیی و، دەست پێگەیشتنیان Access.

```
begin procedure pop: stack
  if stack is empty
    return null
  endif
  data ← stack[top]
  top ← top - 1
  return data
end procedure
```

هەنگاوەکان و کردارەکان بۆ وەرگرتنەوە و دەرهێنانەوەی دانەیەك لە ناو شارای ڕەفەیی Stack ئەمانەی لای خوارەوەن:

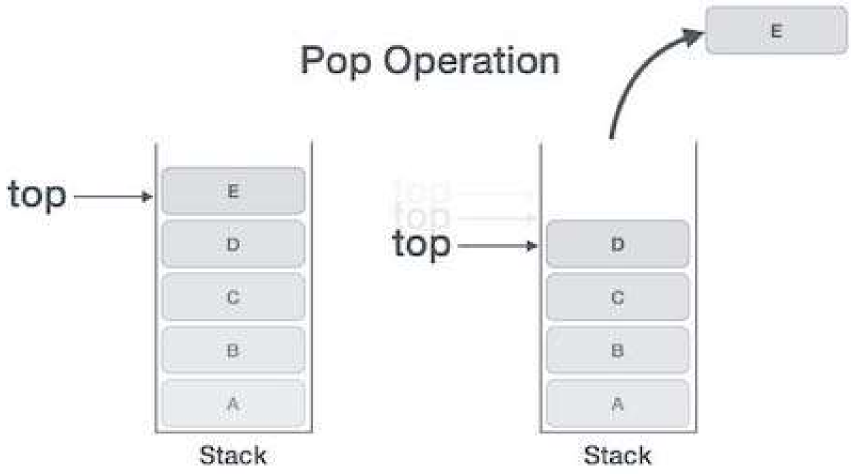
هەنگاوی 1: پشکنینی شارای ڕەفەیی ئەگەر پڕبێت Check If Stack is Full.

هەنگاوی 2: ئەگەر شارای ڕەفەیی پڕبێت، هەڵە Error ڕوودەدات و دەرچوون Exit دەبێت.

هەنگاوی3:ئەگەر شارای ڕەفەیی Stack پڕ نەبێت، دەستگەیشتن بە دانەی زانیاری کە دانەی سەرەوەیە Top.

هەنگاوی4: کەمکردنەوەی نرخی دانەی سەرەوە Top بە یەك (1) دانە.

هەنگاوی5: بەسەرکەوتووانە کارەکە جێبەجێدەبێت.



کرداری گەیشتنە لوتکە Peak () Operation

گەیشتنە پەری ئەو پەر ()Peak: ئەم کردارە بەکاردێت بۆ گەراندنەوەی نرخی لوتکە ((پۆپە)) Peak ی شارای ڕەفەیی، واتە ئەو دانەیەی دەکەوێتە سەری سەرەوە بێئەوەی لای ببات Not Remove.

بڕوانە ئەلگۆریسمەکە:

```
begin procedure peek
  return stack[top]
end procedure
```

کرداری پڕە isFull() Operation

پشکنینی حەجمی شارای ڕەفەیی بۆ پڕێتی ()isFull: شارای ڕەفەی دەپشکنێت ئەگەر پڕ بێت. بڕوانە ئەلگۆریسمی کردارەکە:

```
begin procedure isfull
  if top equals to MAXSIZE
    return true
  else
```

```
      return false
   endif
end procedure
```

isEmpty() Operation کرداری بەتاڵە

پشکنینی حەجمی شارای ڕەفەیی بۆ بەتاڵی isEmpty(): شارای ڕەفەی دەپشکنێت ئەگەر بەتاڵ بێت. بڕوانە ئەلگۆریسمەکە:

```
begin procedure isempty
   if top less than 1
      return true
   else
      return false
   endif
end procedure
```

حاڵەتی شارای ڕەفەیی

Status of Stack

ئەم خشتەیەی خوارەوە حاڵەتی شارای ڕەفەیی Stack Status ڕوونەدەکاتەوە، بە هۆی شوێنگەی سەر Position of Top ەوە:

شوێنگەی سەرە **Position of Top**	حاڵەتی شارای ڕەفەیی **Status of Stack**
-1	شارای ڕەفەیی بەتاڵە. Stack is Empty
0	تەنها یەك دانە لە شارای ڕەفەیی دا هەیە. Only one element in Stack
N-1	شارای ڕەفەیی پڕە. Stack is Full
N	حاڵەتی سەر ریژبوونی شارای ڕەفەیی. Overflow state of Stack

ئاڵۆزێتی کات

Time Complexity

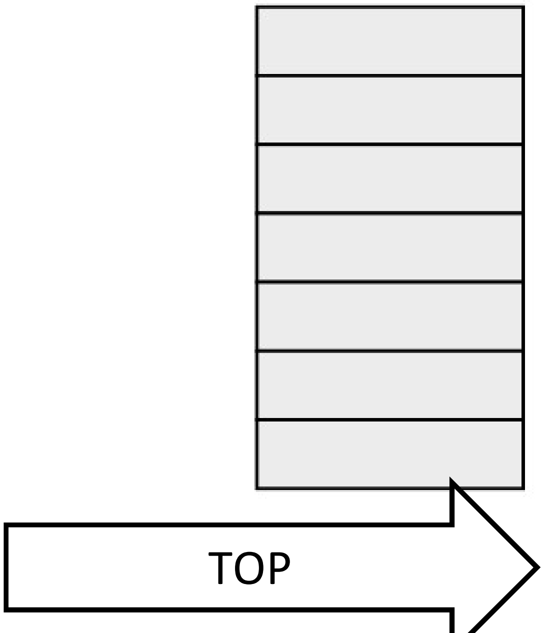
ئاڵۆزێتی کات Time Complexity بە گوێرەی کردارە جیاوازەکان دەبێت، کە جێبەجێدەبێت لەسەر شارای ڕەفەیی Stack، واتە کاتەکە دەگۆڕێت بە گوێرەی کردارەکان، ئەمانەی لای خوارەوە کردارەکانن:

- کرداری خستنەناوی Push Operation: O(1).
- کرداری دەرهێنانەوە(سرینەوە) Pop Operation: O(1).
- کرداری سەرە ((یەکەم دانە)) Top Operation: O(1).
- کرداری گەڕان Search Operation: O(N).

نموونە Example

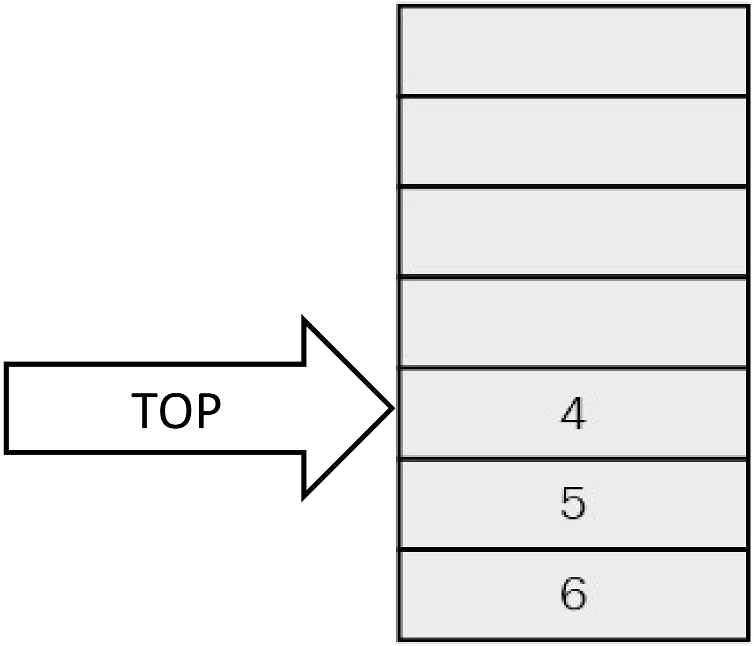
یەکەم// کاتێك شارای ڕەفەیی بەتاڵە:

کاتێك شارای ڕەفەیی Stack بەتاڵە، واتە هیچ دانەیەکی تێدانییە و، لەم کاتەدا حاڵەتی دانەی سەرەوە TOP Element دەکاتە 1-:

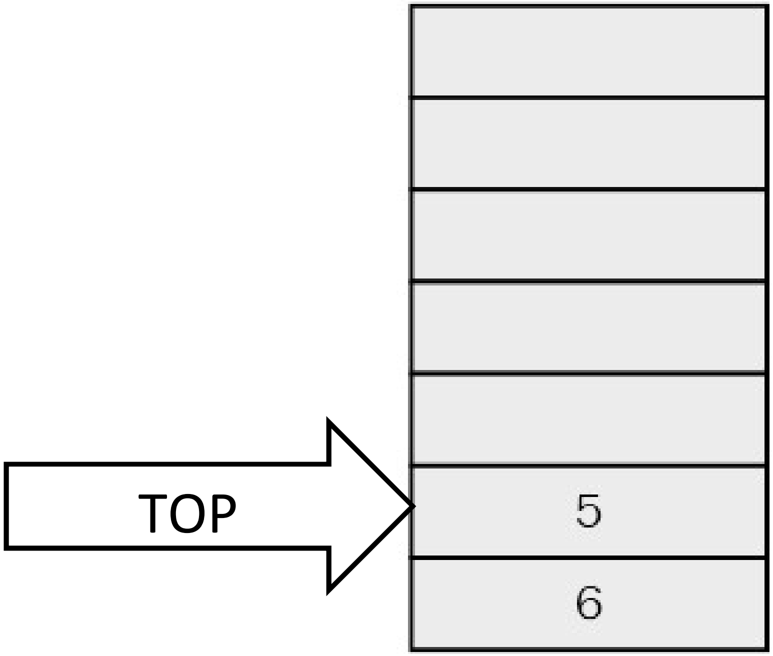


دووەم// کاتێك شارای ڕەفەیی Stack بەتاڵ نییە:

کاتێك ئێکە دانەیەك زیاد دەکەین بۆ ناو شارای ڕەفەیی، ئەوا دانەی سەرەوە بە نرخی 1 زیاد دەکات، دووای زیاد کردن، یەکەم دانەی سەرەوە TOP Element دەکاتە 0.



سێهەم//کاتێك دانەیەکی سەرەوە دەسرینەوە، سەرەوە بە 1 دانە کەم دەکاتەوە:



هێڵکاریی

شێوەی پێکهاتنی بەرنامە

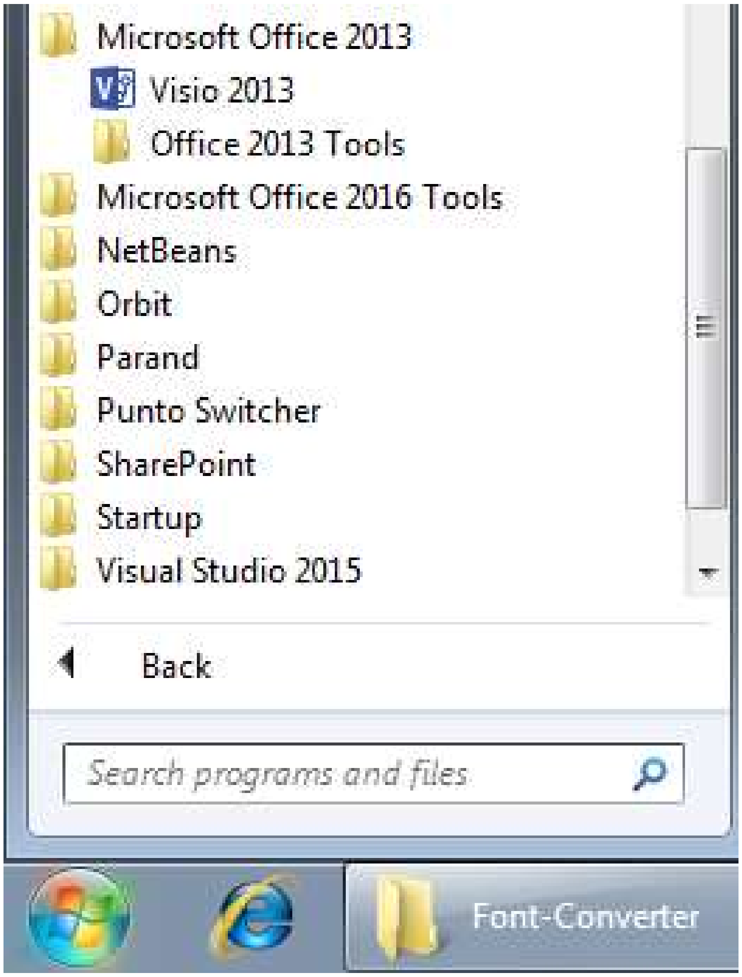
شارای ڕەفەیی

Program Structure Diagram

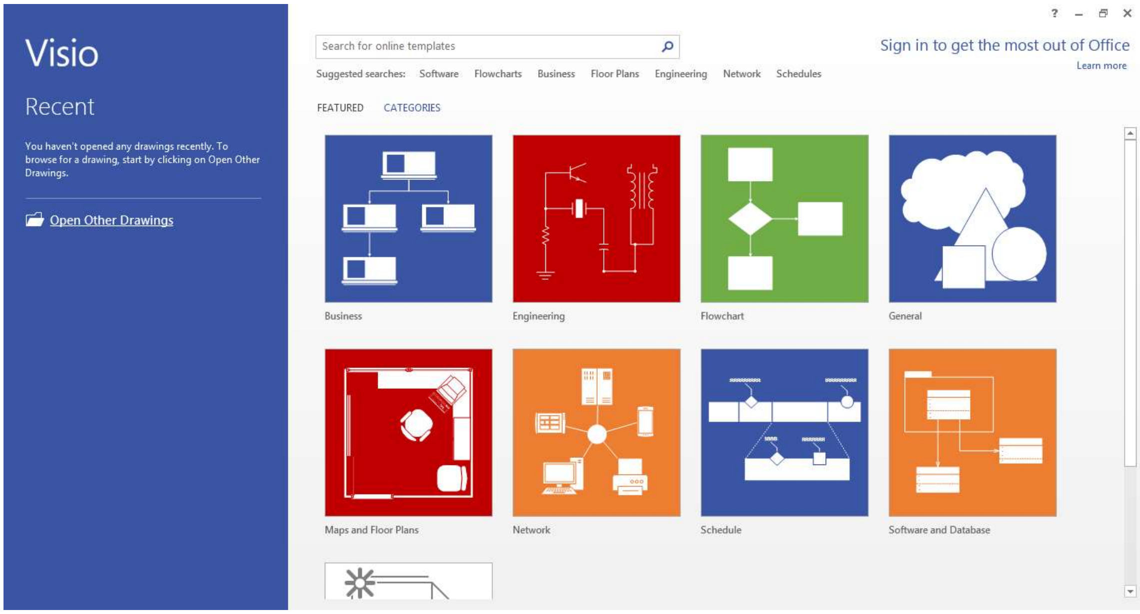
Stack

هەمیشە بەرنامەڕێژیی و پێویستی دیاری کردن و نەخشە سازیی لە دیارترین هەنگاوەکانی کاری بوواری تەکنەلۆژین، یەکێك لەو هێڵکارییانەش کە بۆ وێنەکێشی ڕوونکردنەوەی شێوەی پێکهاتنی زانیاری و، خستنە ڕووی ((شارای ڕەفەیی Stack)) بەکاردەهێنرێت، ناوی ((هێڵکاری شێوەی پێکهاتنی بەرنامەیە Program Structure)) کە لەبەشی بنکەی زانیاری و بەرنامەی Software and Database بەرنامەی مایکرۆسۆفت ڤیزیۆ 2016 دانراوە، بۆیە بە کورتی باسی بەکارهێنانی دەکەین لە نموونەیەك دا.

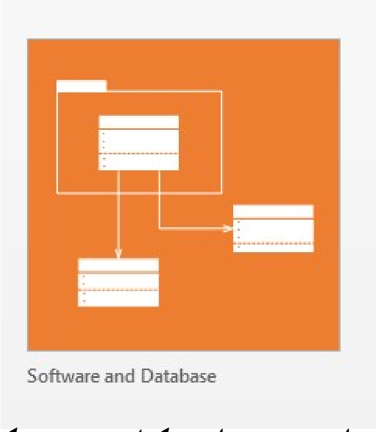
1-بەرنامەی مایکرۆسۆفت ڤیزیۆ 2016یان 2013 دەکەینەوە، بە کلیك کردن لەسەر دووگمەی دەستپێکردن Start و، پاشان کلیك کردن لەسەر هەموو بەرنامەکان All Program، دوواتر کلیك کردن لەسەر مایکرۆسۆفت ئۆفیس 2013، لە کۆتایدا کلیك کردن لەسەر مایکرۆسۆت ڤیزیۆ 2013:



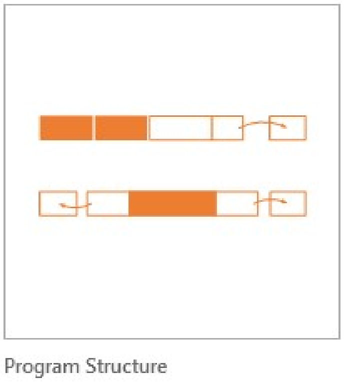
2-ئیستا بەرنامەکە کراوەتەوە:



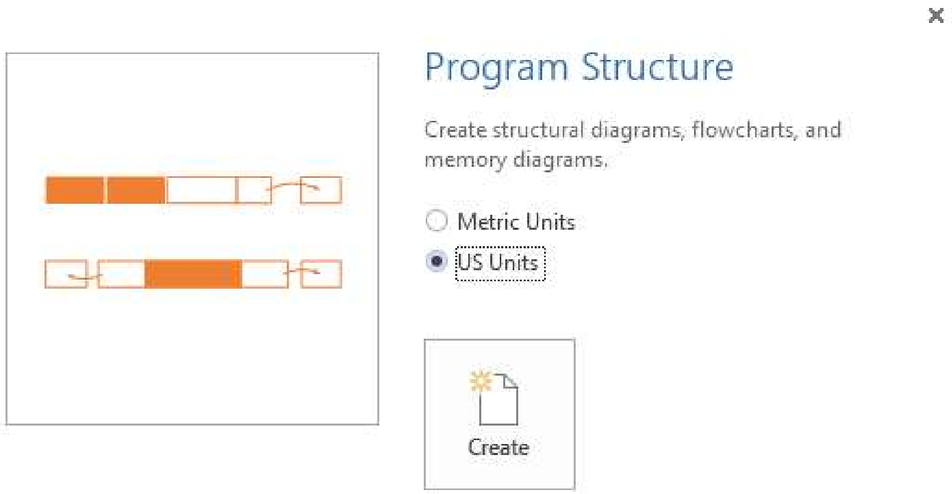
3- كلیك لەسەر پۆلێنی بنکەی زانیاری و بەرنامە Software and Database دەکەین:



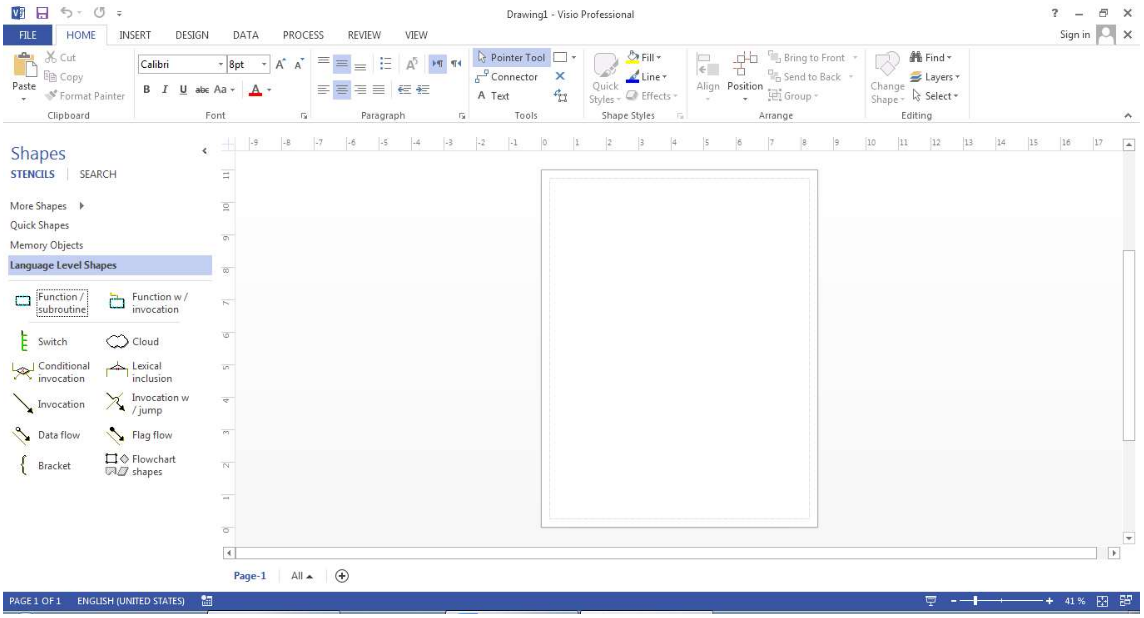
4-ئیستا بەشی بنکە ی زانیاری و بەرنامە کراوەتەوە و، کلیك لەسەر هێڵکاری شیوەی پێکهاتنی بەرنامە Program Structure Diagram، دەکەین:



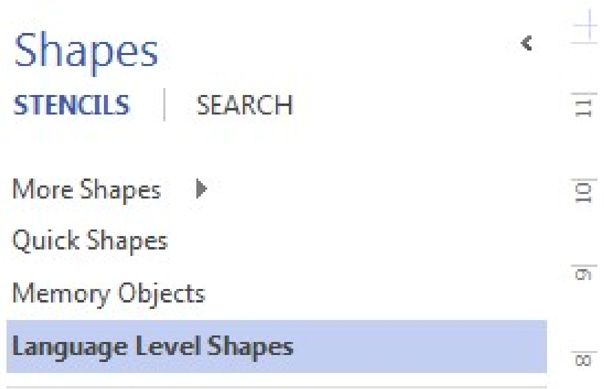
5-كلیك لەسەر دووگمەی درووستکردن Createدەکەین:



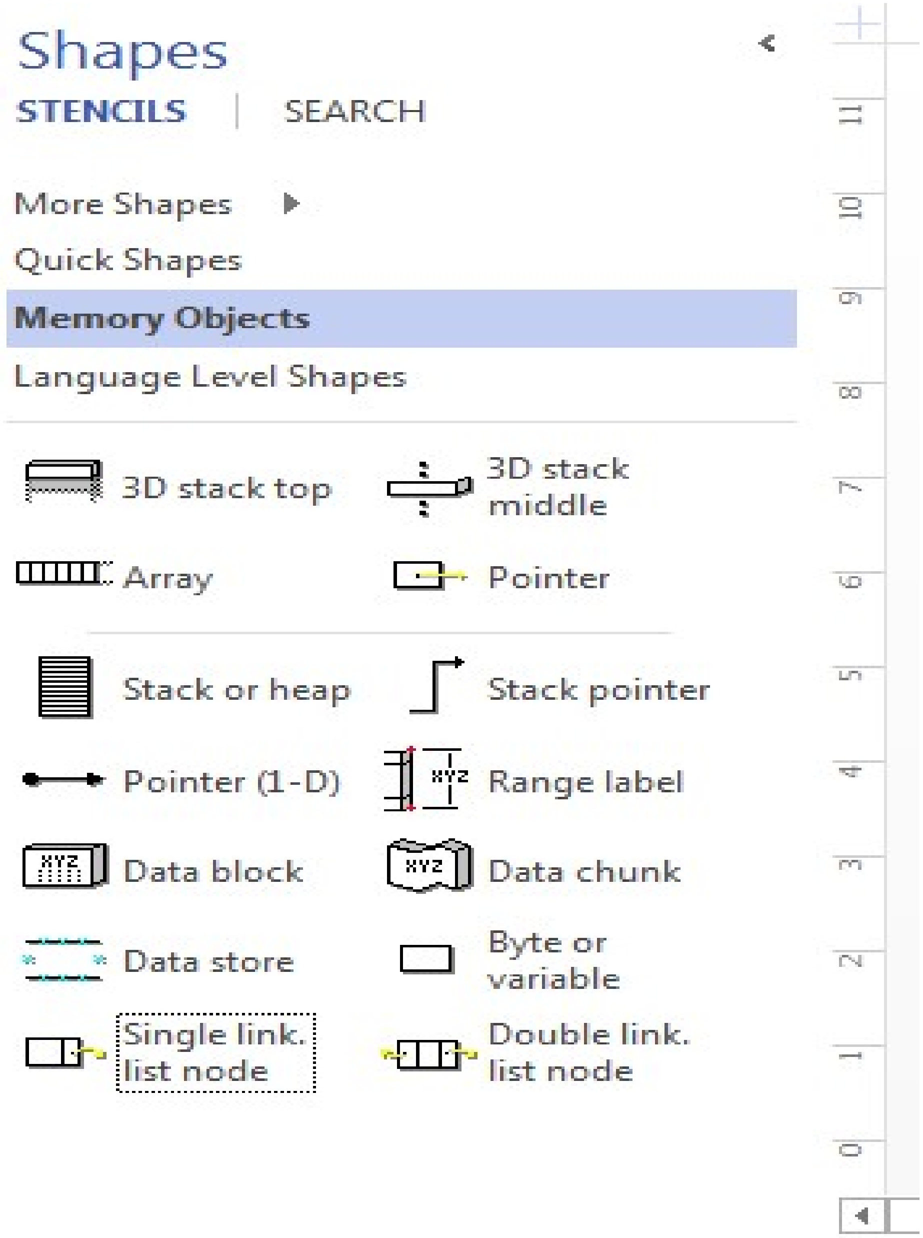
6-هیلکارییەکە چالاکە بۆ کێشانی شارای ڕەفەیی Stack:



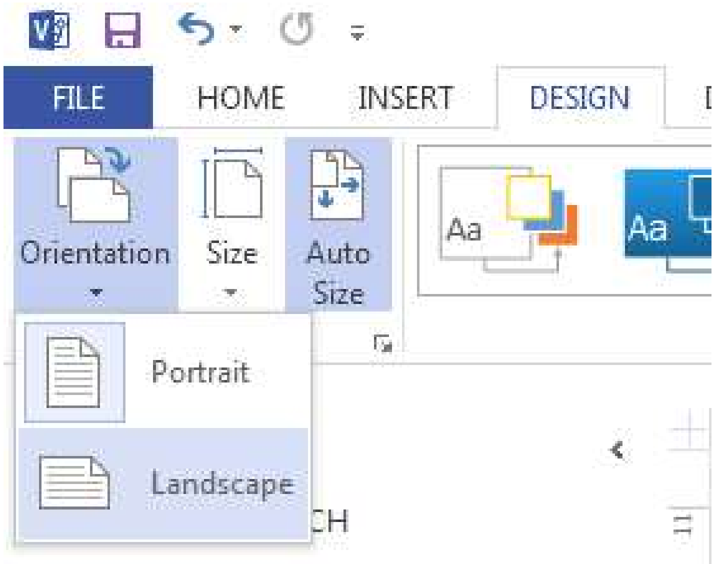
7-لە بەشی دەستەچەپی بەرنامەکە و لە ژێر شێوەکان Shapesدا، کلیك لەسەر تەنەکانی میمۆری Memory Objectsدەکەین:



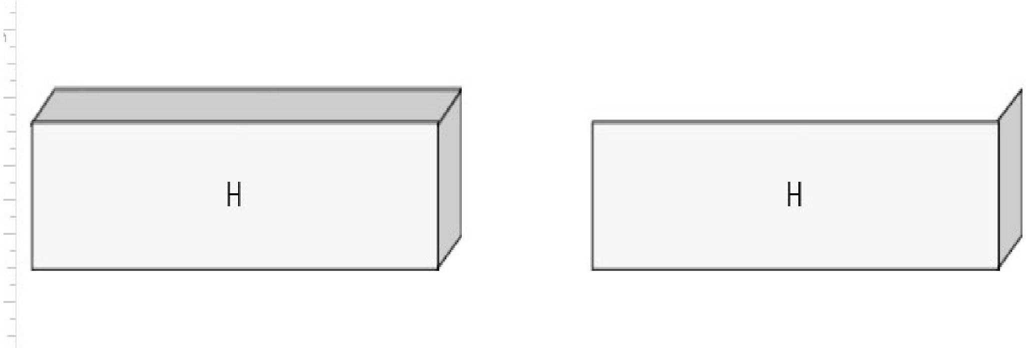
8-بەشەکانی تایبەت بە تەنەکانی میمۆری Memory Objects دەردەکەوێت و، هەر شێوەیەکان ویست هەڵیدەگرین و رای دەکێشینە سەر لاپەرەی کار :



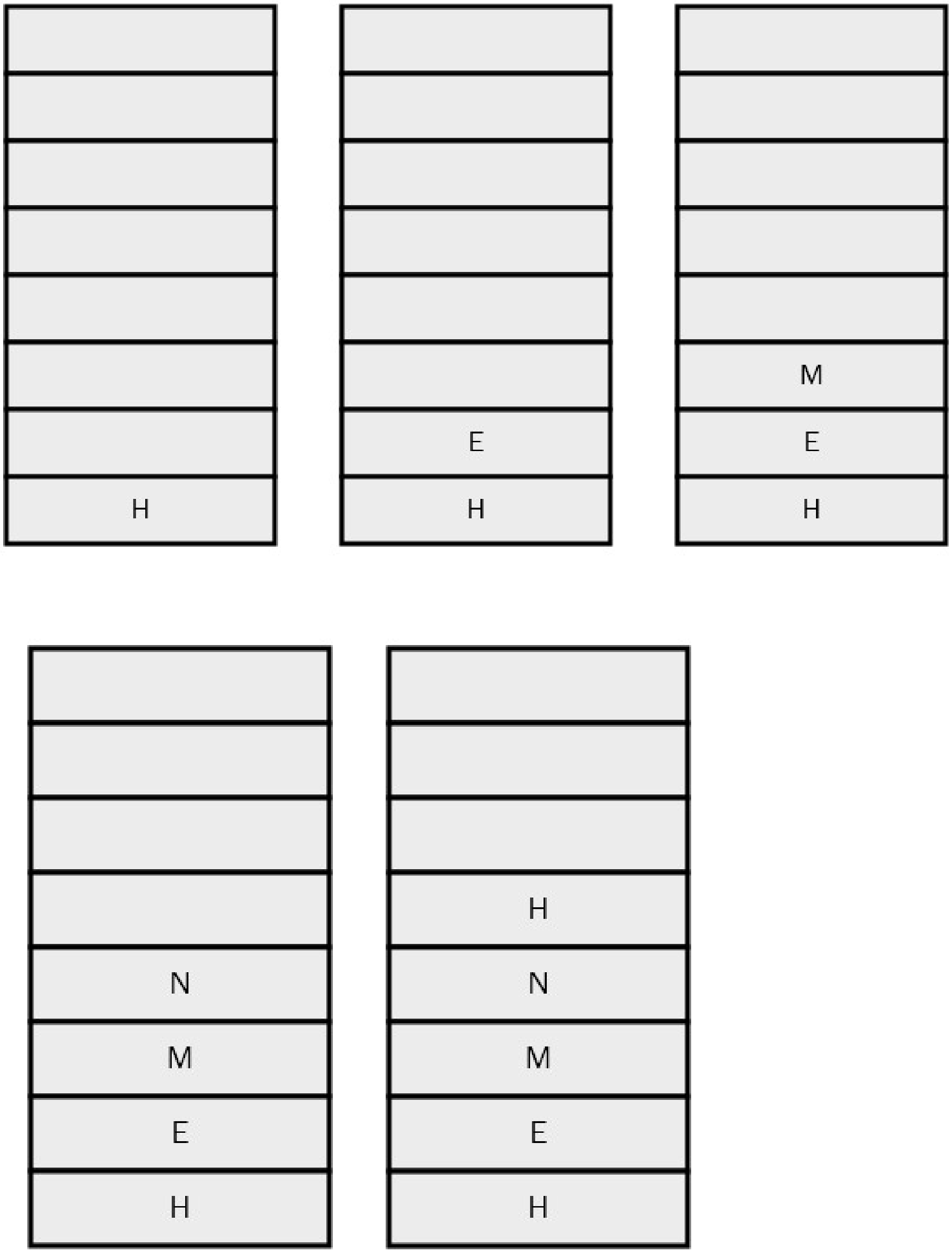
9-ئەگەر هێڵكارییەكەمان لە باری درێژیی بێت ئەوا واباشترە لاپەڕەكە بخەینە باری درێژیی، بە كلیك كردن لە سەر تابی نەخشەسازیی Design، پاشان كلیك كردن لەسەر ئاراستە Orientation، و دوواتر كلیك كردن لەسەر Landscape:



10-ئیستا دەتوانین سوود لە شێوەی سێ ڕەهەندی تۆپ 3D Stack Top وەربگرین بۆ كێشانی یەك دانەی سێ ڕەهەندی، یاخود شێوەی سێ ڕەهەندی ناوەراست 3D Stack Middle، بە راكێشانی هەر یەكێكیان بۆ سەر لاپەڕەی كێشانی هێڵكاری و نووسینی زانیاری تیایدا.



11-ئەگەر بمانەوێت ((شارای ڕەفەیی Stack)) بە تەواوی بكێشین، ئەوا شێوەی ستاك رادەكێشینە سەر لاپەڕەی هێڵكاریی و، دانەكانی بۆ داغڵ دەكەین، یاخود بە وێنەی تر لە تەنیشتییەوە دانەكانی لێدەر دەكەینەوە:


H
E
H
M
E
H
N
M
E
H
H
N
M
E
H

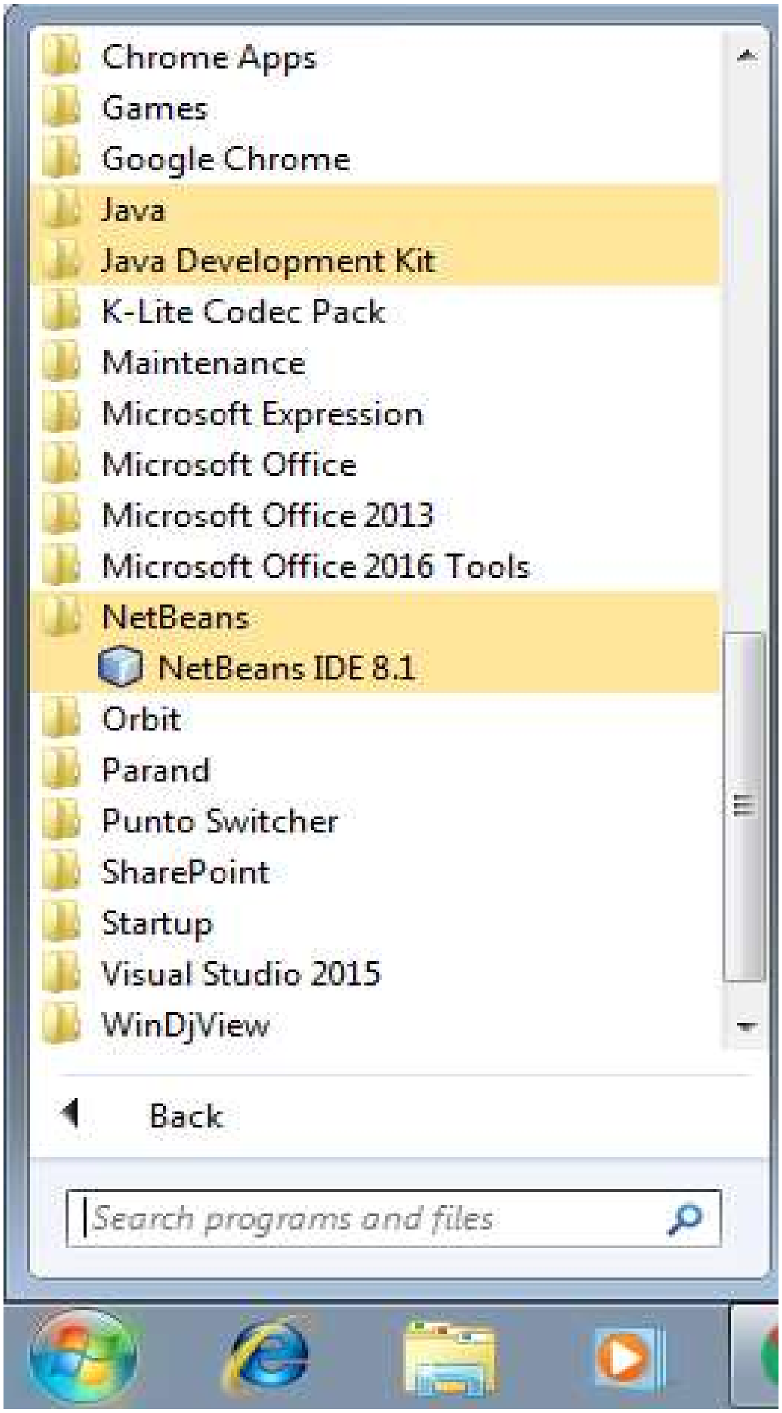
جێبەجێکردنی شارای ڕەفەیی

Stack Implementation

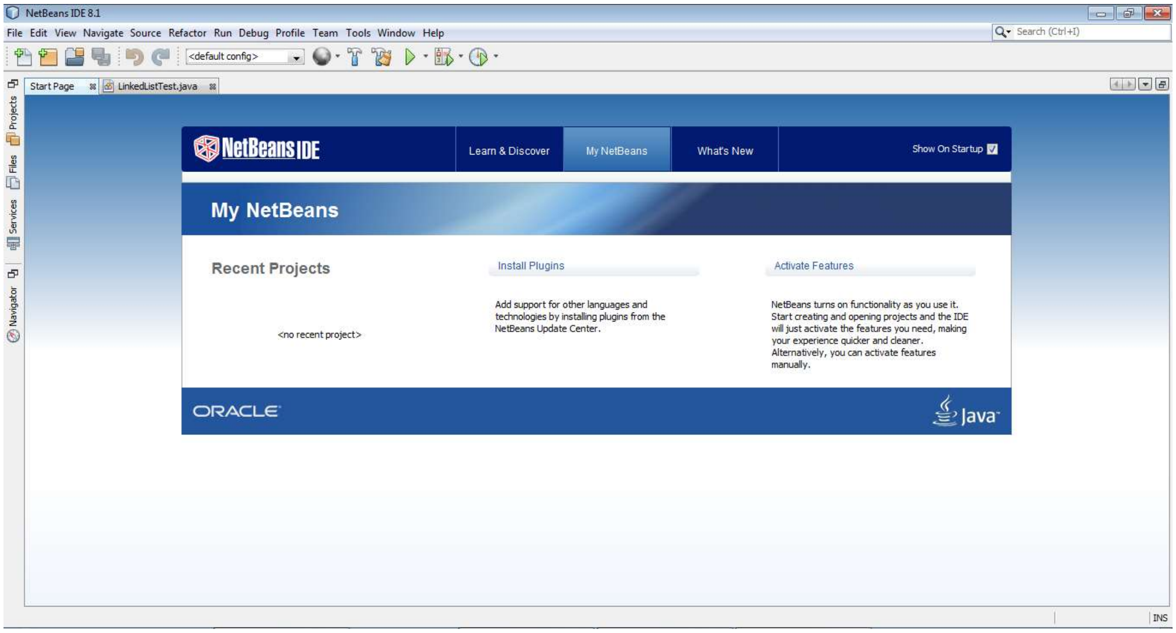
جێبەجێکردنی شارای ڕەفەیی

Stack Implementation

1-بەرنامە نێتبین NetBeans دەکەینەوە، بە کلیك کردن لەسەر دووگمەی دەستپێکردن Start Button، پاشان کلیك لەسەر NetBeans و دوواتر کلیك لەسەر NetBeans IDE 8.1:

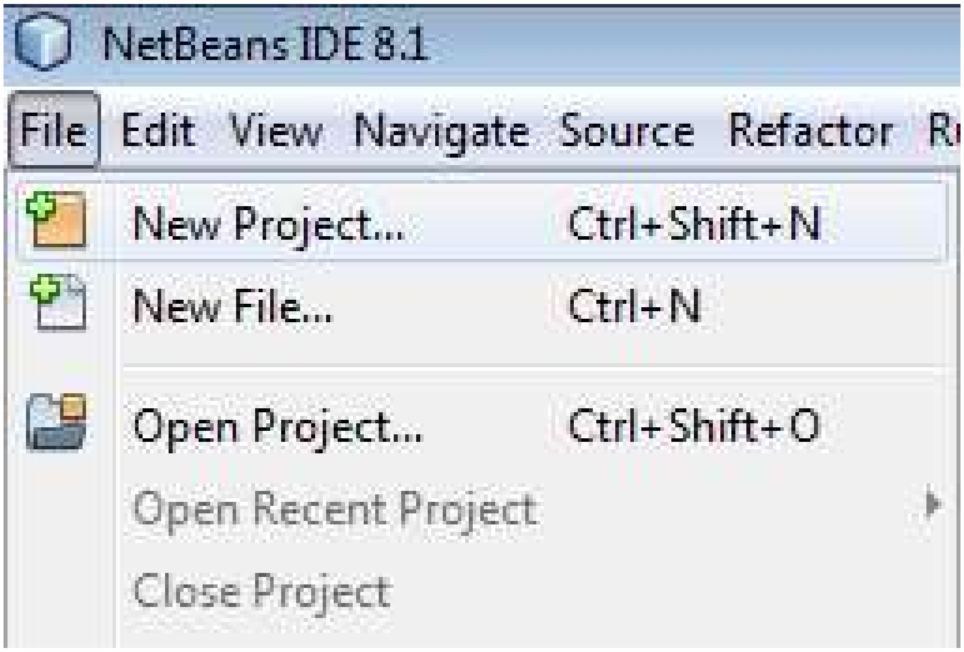


2-ڕووکاری بەرنامەی نێتبینز NetBeansدەکرێتەوە.

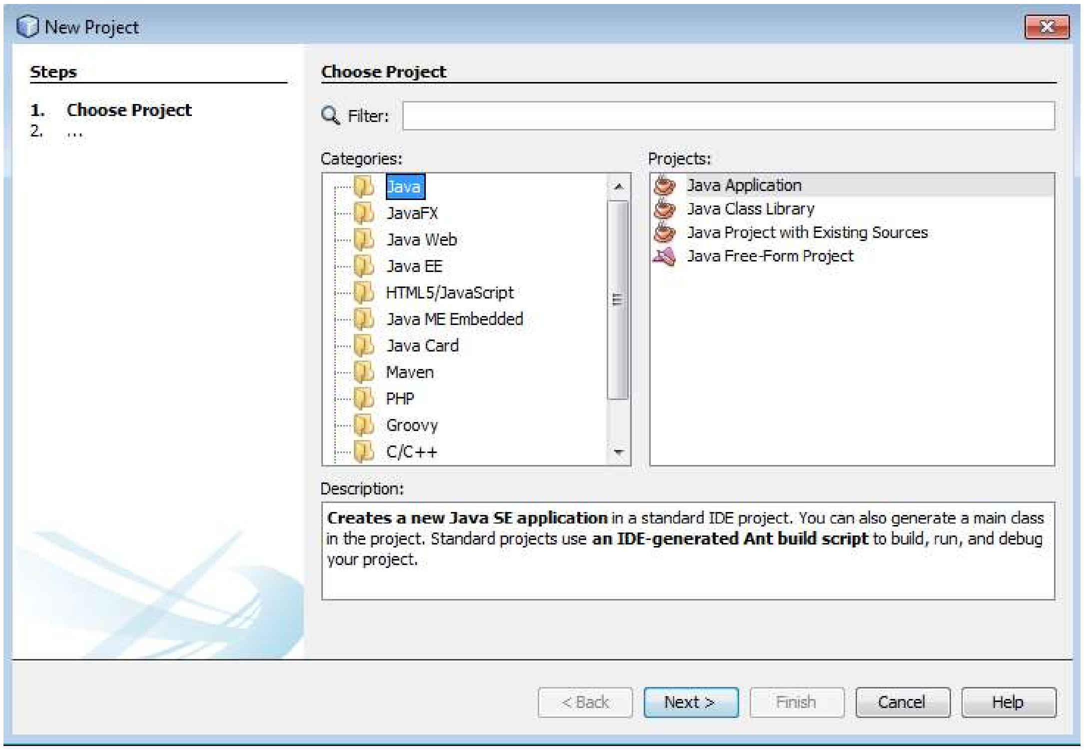


3-کلیك لەسەر دووگمەی فایل Fileدەکەین، بۆ ئەوەی پرۆژەیەکی نوێ درووست بکەین.

4-کلیك لەسەر پرۆژەی نوێ New Project دەکەین:



5-ڕووکاری تایبەت بە پرۆژەی نوێ New Projectدەکرێتەوە و، کلیك لەسەر دووگمەی دوواتر Next دەکەین.



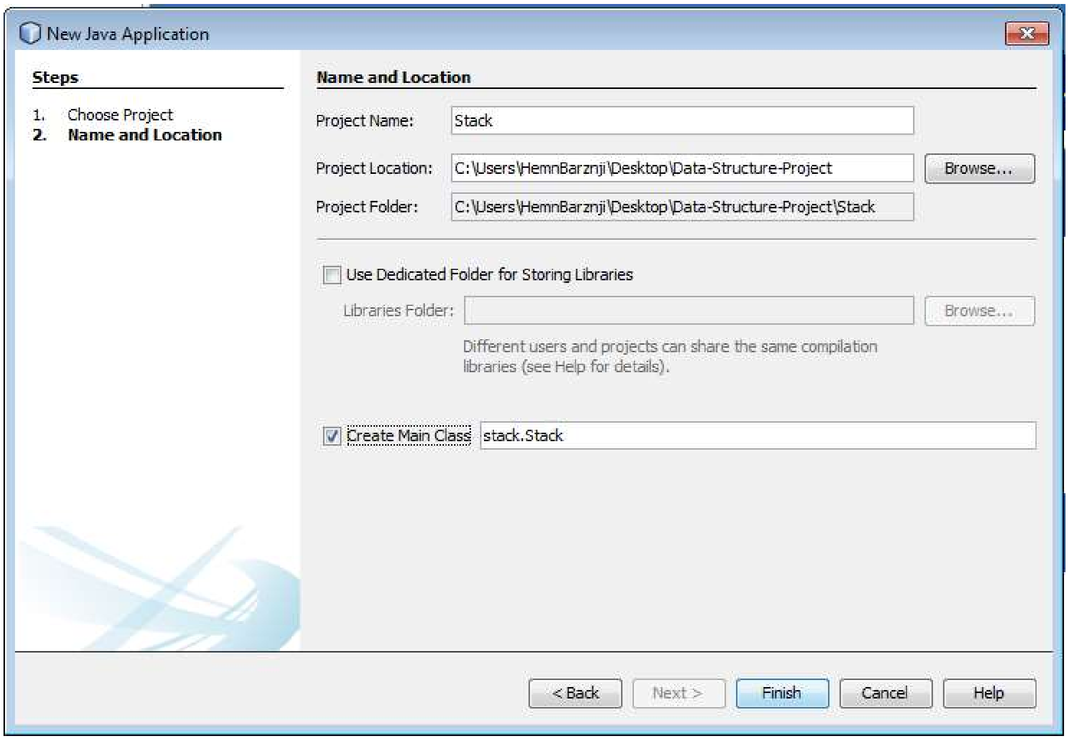
6-ڕووکارێکی نوێ دەکرێتەوە:

-لەبەشی ناوی پرۆژە Project Nameدا، ناوێك بۆ پرۆژەکە بنووسە.

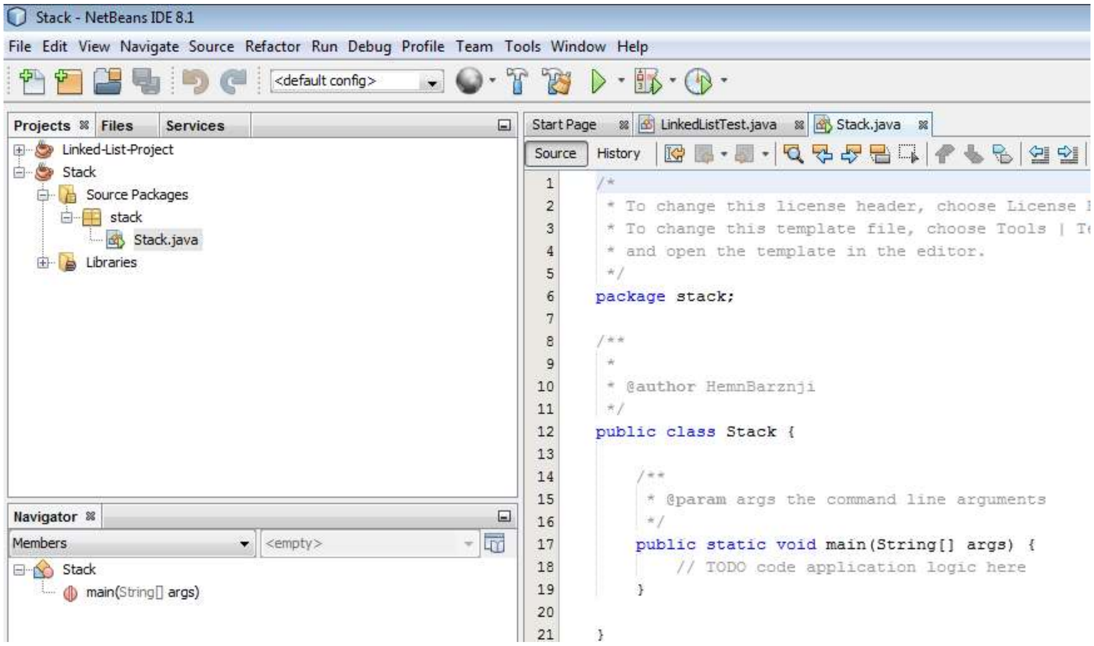
-لەبەشی شوێنی پرۆژە Project Locationدا، شوێنی پرۆژەکە دیاری بکە.

-کلیك لەناو سنووقی پشکنینی Check Box درووستکردنی پۆلی سەرەکی Create Main Class بکە، ئەگەر دەتەوێت پۆلێکت بۆ درووست بکات بە ناوێکەوە کە بەرنامەکە دیاری دەکات.

-لە دووای ئەو هەنگاوانە کلیك لەسەر دووگەی کۆتای هاتن Finishبکە.



7-ئیستا پرۆژەکە درووست بوو:



8-بەمەش پۆلێك Class درووست بوو، کە ڕێگەی سەرەکیشی Main Class تیایە.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */

package stack;

/**
* @author HemnBarznji
 */
public class Stack {
   public static void main(String[] args) {
       // TODO code application logic here
   }
}
```

كۆد نووسی بۆ شارای ڕەفەیی

Coding for Stack

1-سەرەتا پاکەتی جاڤا یوتیلْ Java Util دەهێنینە ناوەوە:

```
import  java.util.*;
```

2-ڕاگەیاندن و بڵاوکردنەوەی شارای ڕەفەیی Stack Declaration ئەنجام دەدەین، لەناو ڕێگەیی سەرەکی Main Method دا، کە وشەکلیلی Stack دەنووسین و، ئەمەش ناوی پۆلی Class شارای ڕەفەییەStack و، بۆشاییەك دادەنێین، ئینجا ناوێك وەك تەن Object دەنووسین، کە دەبێتە ناوی شارای ڕەفەیی Stack و، تەنێکە لە پۆلی شارای ڕەفەیی Object of Stack، و پاشان هێمای = و وشەکلیی نویْ New و، بەهەمان شێوە ناوی پۆلی شارای ڕەفەیی و و جووت کەوانەی کۆتایی و فاریزەیی نوقتەدار لە دووایدا بڕوانە کۆدەکەی خوارەوە:

```
// creating stack
  Stack st = new Stack();
```

3-پاشان بە هۆیی ڕێگەیی زیاد کردنەوە PUSH Method دانەکان زیاد دەکەین بۆ ناو شارای ڕەفەیی Stack، کە سەرەتا ناوی شارای ڕەفەیی Stack دەنووسین و پاشان نووقتە دادەنێین و وشەی زیاد کردن push. بە دووایدا دەنووسین و لە نێوان کەوانە و دووانە کۆمای سەرەوەدا (“ ”) ئەو نرخە دەنووسین کە دەمانەوێت داغڵی بکەین:

```
        st.push("Hemn");
```

4-بەهەمان شێوە هەموو دانەکانی تر زیاد دەکەین کە دەمانەوێت:

```
  st.push("Barznji");
  st.push("May 1-1982");
```

5-ڕێگەی پیشاندان Display Method دەنووسین، بۆ چاپکردن:

```
System.out.println("stack: " + st);
```

6-ڕێگەی بەتاڵ ()empty بەکاردەهێنین بۆ پشکنینی شارای ڕەفەیی:

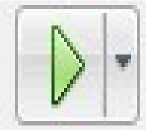
```
// checking stack
  System.out.println("Is stack empty: "+st.empty());
```

ئەمەش هەموو کۆدەکەیە پێکەوە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */


/**
* @author HemnBarznji
 */
import java.util.*;
public class Stack_DS_Book {
    public static void main(String[] args) {
        // creating stack
  Stack st = new Stack();
```

```java
   // populating stack
   st.push("Hemn");
   st.push("Barznji");
   st.push("May 1-1982");
   // checking stack
   System.out.println("Is stack empty: "+st.empty());
      System.out.println("stack: " + st);
    }
}
```

7-کلیك لەسەر دووگمەی جیبەجێکردنی پرۆژە Run Project دەکەین.



8-ئەنجام بەم جۆرە دەبێت:

```
Is stack empty: false
stack: [Hemn, Barznji, May 1-1982]
```

ڕێگەکان Methods

ڕێگەی بەتاڵ

Empty () Method

ڕێگەی بەتاڵ Empty() Method ، بەکاردەهێنرێت بۆ تاقیکردنەوەی شارای ڕەفەیی Stack، کە لە نموونەکەی سەرەوەدا بەکارهاتووە، نرخی راست True دەگێرێتەوە، ئەگەر شارای ڕەفەیی Stack بەتاڵ بێت، بە پێچەوانەوە نرخی هەڵە False دەگێرێتەوە. بۆ نموونە ئەگەر دووای درووستکردنی تەن لە پۆلی شارای ڕەفەیی Stack و، پێش داغڵکردنی زانیاری بۆ شارای ڕەفەیی Push Element، ئەم ڕێگەیە بەکاربهێنین نرخی راست True دەگەرێنێتەوە.

```
// creating stack
  Stack st = new Stack();
  // checking stack
  System.out.println("Is stack empty: "+st.empty());
```

ڕێگەی خستنە ناو Push() Method

خستنە ناو ((پاڵنان بۆ ناو)) () Push: ئەم ڕێگەیە بەکاردێت بۆ زانیاری خستنە ناو شارای ڕەفەیی Stack و، پاشەکەوت کردنی Store زانیاریی. واتە دانەکان دەخاتە ناو شارای ڕەفەیی. ناوی تەنی Object شارای رەفەیی و نوقتە و رێگەی push بەکار دەهێنین و لە نێوان دوو کەوانەی داغڵکەری ڕێگەکە ئەو نرخە دەنووسین کە دەمانەوێت داغڵی بکەین لە نێوانە دووانە فاریزەی سەرەوە و لە کۆتایشدا فاریزەی نووقتەدار دادەنێین:

```
  st.push("Hemn");
```

بڕوانە نموونەکەی لای خوارەوە کە بەرنامەیەکی تەواوە.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
/**   @author HemnBarznji   */
import java.util.*;
public class Stack_DS_Book {
   public static void main(String[] args) {

       // creating stack
   Stack st = new Stack();
   // checking stack
   System.out.println("Is stack empty: "+st.empty());
   // populating stack
   st.push("Hemn");
   st.push("Barznji");
   st.push("May 1-1982");
      System.out.println("stack: " + st);
```

```
    }
}
```

ئەنجام//

Is stack empty: true

stack: [Hemn, Barznji, May 1-1982]

ڕێگەی گەیشتنە لوتکە Peak () Method

گەیشتنە پەری ئەو پەر Peak(): ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی نرخی لوتکە ((پۆپە)) Peak ی شارای ڕەفەیی، واتە ئەو دانەیەی دەکەوێتە سەری سەرەوە بێئەوەی لای ببات Not Remove.

```
/*
*Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
*/

/**   * @author HemnBarznji       */
import java.util.*;
public class Stack_DS_Book {
    public static void main(String[] args) {
        // creating stack
   Stack st = new Stack();
```

```
  // checking stack
  System.out.println("Is stack empty: "+st.empty());

  // populating stack
  st.push("Hemn");
  st.push("Barznji");
  st.push("May 1-1982");
     System.out.println("stack: " + st);
     // checking the top object
  System.out.println("Top object is: "+st.peek());
   }
}
```

ئەنجام//

```
Is stack empty: true
stack: [Hemn, Barznji, May 1-1982]
Top object is: May 1-1982
```

ڕێگەی دەرهێنانەوە ((لابردن)) POP() Method

دەرهێنانەوە ()Pop: ئەم ڕێگەیە بەکاردێت بۆ لابردنەوەی Remove دانەکانی ناو شارای ڕەفەیی و، دەست پێگەیشتنیان Access. کە یەکەم دانەی سەرەوە لا دەبات. . ناوی تەنی Object شارای ڕەفەیی و نوقتە و ڕێگەی pop بەکار دەهێنین و لە نێوان دوو کەوانەی داغڵکەری ڕێگەکە هیچ نرخێك نانووسین و خۆیی یەکە دانەی سەرەوە TOP لا دەبات و لە کۆتایشدا فاریزەی نووقتەدار دادەنێین:

```
st.pop()
```

بڕوانە نموونەکەی لای خوارەوە کە بەرنامەیەکی تەواوە.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor

 */

/**   @author HemnBarznji   */
import java.util.*;
public class Stack_DS_Book {
    public static void main(String[] args) {

        // creating stack
    Stack st = new Stack();



    // checking stack
    System.out.println("Is stack empty: "+st.empty());
```

```
// populating stack
st.push("Hemn");
st.push("Barznji");
st.push("May 1-1982");

System.out.println("stack: " + st);
// removing top object
System.out.println("Removed object is: "+st.pop());
// elements after remove
System.out.println("Elements after remove: "+st);
}
}
```

ئەنجام//

```
Is stack empty: true
stack: [Hemn, Barznji, May 1-1982]
Removed object is: May 1-1982
Elements after remove: [Hemn, Barznji]
```

ڕێگەی گەڕان

Search () Method

ئەم ڕێگەیە بەکاردێت بۆ گەڕان بە دووای دانەیەکی ناو شارای ڕەفەیی Stack، کە دانەکە لەناو کەوانەی ڕێگەی گەڕان دا دەنووسین:

```
st.search("Barznji");
```

بڕوانە هەموو کۆدەکە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */


/**   @author HemnBarznji   */
import java.util.*;
public class Stack_DS_Book {
    public static void main(String[] args) {


        // creating stack
   Stack st = new Stack();
```

```
  // checking stack
  System.out.println("Is stack empty: "+st.empty());

  // populating stack
  st.push("Hemn");
  st.push("Barznji");
  st.push("May 1-1982");

     System.out.println("stack: " + st);
   // searching 'Barznji' element
System.out.println("Searching 'Barznji' in stack:
"+st.search("Barznji"));
   }
}
```

ئەنجام//

```
Is stack empty: true
stack: [Hemn, Barznji, May 1-1982]
Searching 'Barznji' in stack: 2
```

نموونەی شارای ڕەفەیی بۆ ژمارەی تەواو

Stack Example for Integer

لەم نموونەیەدا، ڕێگە ئامادەکراوەکان لە ناو ڕێگەی تر دا بەکارهاتووە کە خۆمان درووستمان کردووە و تایبەتە بە ژمارەی تەواو و دەتوانین بەم جۆش شارای ڕەفەیی Stackدرووست بکەین و تاقیبکەینەوە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
/**   @author HemnBarznji   */
import java.util.*;
public class Stack_DS_Book {



    static void showpush(Stack st, int a) {
     st.push(new Integer(a));
     System.out.println("push(" + a + ")");
     System.out.println("stack: " + st);
  }

  static void showpop(Stack st) {
```

```
    System.out.print("pop -> ");
    Integer a = (Integer) st.pop();
    System.out.println(a);
    System.out.println("stack: " + st);
  }

  public static void main(String[] args) {

Stack st = new Stack();
    System.out.println("stack: " + st);
    showpush(st, 42);
    showpush(st, 66);
    showpush(st, 99);
    showpop(st);
    showpop(st);
    showpop(st);
    try {
      showpop(st);
    } catch (EmptyStackException e) {
      System.out.println("empty stack");
    }
```

```
    }
}
```

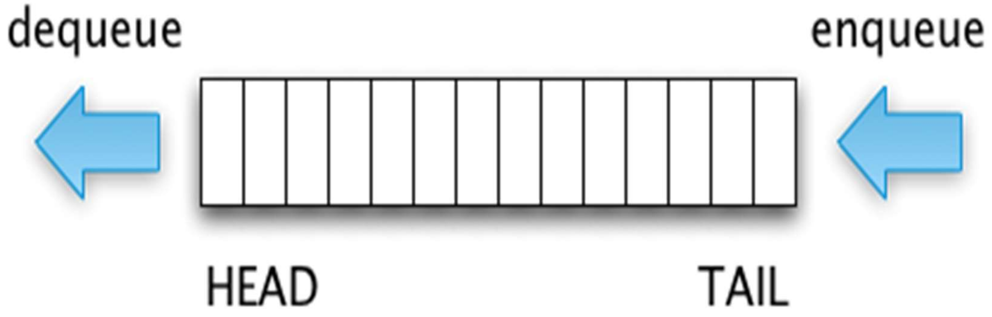
ئەنجام//

```
stack: []
push(42)
stack: [42]
push(66)
stack: [42, 66]
push(99)
stack: [42, 66, 99]
pop -> 99
stack: [42, 66]
pop -> 66
stack: [42]
pop -> 42
stack: []
pop -> empty stack
```

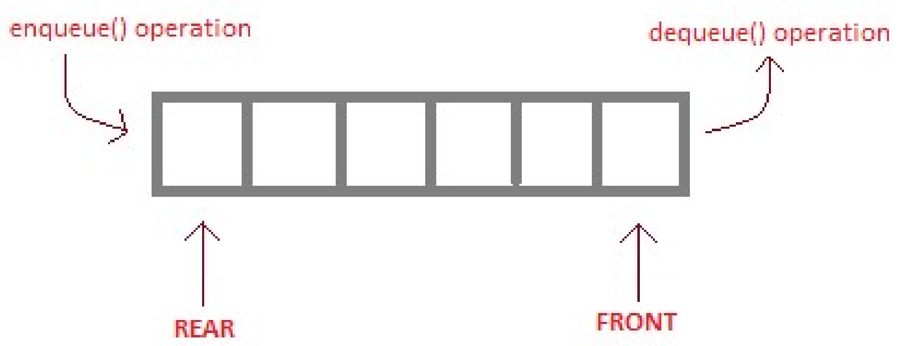
شارای

نۆبە بەستن

Queue



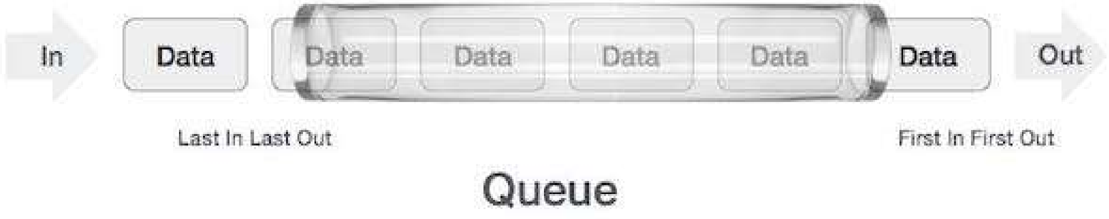
نۆبە بەستن ((ریز بەستن)) Queue جۆری زانیاری پوختەیە ((ڕووتە)) Abstract Data Type، و دەتوانین بلێین وەك شارای ڕەفەیی Stack وایە، بەڵام لە هەردوو لاوە کراوەتەوە، لایەکیان بەکاردەهێنرێت بۆ چونە ناوەوە ی دانەکان بۆ نۆبە بەستن ((نۆبە دەبەستیٚ Enqueue)) و، کە پێی دەڵێن: دوواوە ((پشتەوە Rear))، یان ((کلك Tail))، لایەکەی تری بەکاردێت بۆ لابردن و سرینەوەی دانەکان و چونەدەرەوە لە نۆبەکە ((چونەدەر لە نۆبە Dequeue)) و، پێی دەڵێن : پێشەوە ((بەردەم Front)).



ئەم نۆبە بەستنە بەجۆرێکە، کە یەکەم دانەی داغڵبوو First – In یەکەم دانەی دەرچووش First – Out دەبێت، کرداری زیاد کردنی دانە بۆناو نۆبەکە پێی دەڵێن Enqueue ، بەڵام کرداری لابردن و سرینەوەی دانە لە نۆبەکەدا، پێی دەڵێن: Dequeue.

نموونەی ڕاستەقینە و بەرچاو بۆ تێگەیشتنی نۆبە بەستن Queue، ڕێگەیەکی تاك ئاڕاستەیە کە یەکەم سەیارەی داغڵبوو، یەکەم سەیارەش دەبێت دەردەچێت، و بەو جۆرە:

بەهەمان شێوەی شارای ڕەفەیی Stack، دەتوانین ریزکراوە Array یان لیستەی پێکەوە بەستراوو Linked List بەکاربهێنین بۆ درووستکردنی نۆبە بەستن Queue.



ڕووخسارە بنچینەییەکانی نۆبە بەستن

Basic Features of Queue

1- وەکو شارای رەفەیی Stack، نۆبە بەستن Queue لیستەی رێکخراوە Ordered List ی دانەکانە لە هەمان جۆری زانیاری Similar Data Type.

2- شیوەی نۆبە بەستن Queue، وایە کە یەکەم دانەی داغڵبوو First – In، یەکەم دانەشە دەچێتە دەرەوە First – Out، و بە شێوەی فیفۆ FIFO ناسراوە.

3- کاتێك دانەیەکی نوێ داغڵدەبیت بۆ ناو نۆبە بەستنەکە، پێویستە هەموو دانەکانی ناو نۆبەکە کە پێشتر چونەتە ناو نۆبەکەوە لا ببرێن Remove و بچنە دەرەوە، بۆ لابردنی دانە نوێیەکە.

4- رێگەی گەیشتن بە لوتکە () Peak بەکاردێت بۆ گەراندنەوەی نرخی یەکەم دانە، بێ ئەوەی لابردن و سرینەوە Dequeueئەنجام بدرێت.

جیبەجێکردنەکانی نۆبە بەستن

Applications of Queue

- داواکاری خزمەتگوزاری لە یەك سەرچاوەی هاوبەشەوە Shared Resource، وەك چاپکەر Printer، خشتەی کاری سی پی یوو CPU Task Scheduling، و ..... تاد.

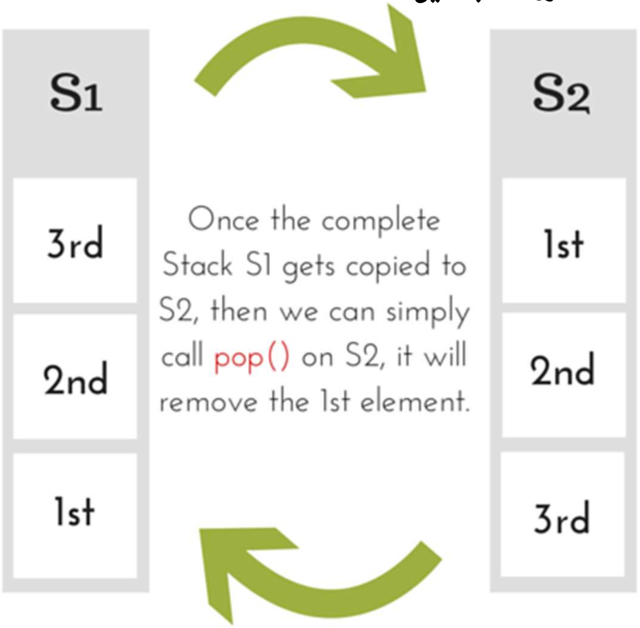
- لە ژیانی رۆژانەدا، سیستەمەکانی سەنتەرەکانی پەیوەندی تەلەفۆنی، نۆبە بەستن Queue بەکاردەهێنن.

- هێڵی سەیارەکان Line of Cars.
- چاوەڕوانی پەکەتەکان Packets Waiting لە راوتەردا Router.
- شێوە سازی ((سیمولەیشن)) Simulation.
- جیاکەرەوەی ((بەفەری)) چاپکەر Printer Buffer.
- زۆر نموونە و بەکارهێنانی تر.

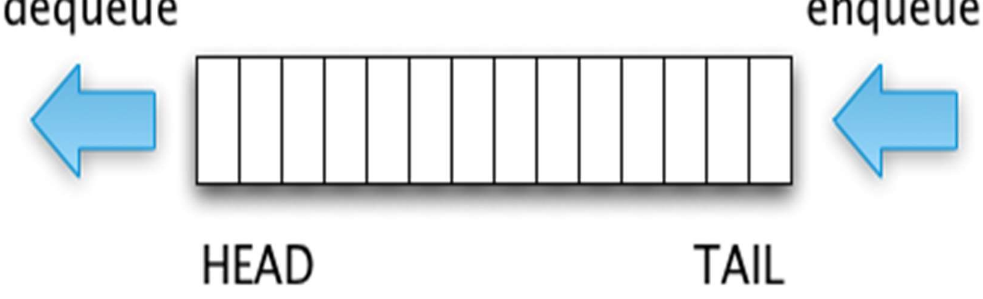
درووستکردنی نۆبە بەستن

Queue Implementation

دەتوانین نۆبە گرتن Queue بە بەکارهێنانی ریزکراوە Array، شارای ڕەفەیی Stack، یان لیستەی پێکەوە بەستراوو Linked List دروست بکەین.



بەڵام ئاسانترینیان بەکارهێنانی ریزکراوە Arrayیە بۆ درووست کردنی نۆبە، کە سەر Head واتە پێشەوە Front دەناسێنین، لەگەڵ دوواوە Rear واتە کلك Tail،



کرداره بنچینەییەکان Basic Operations

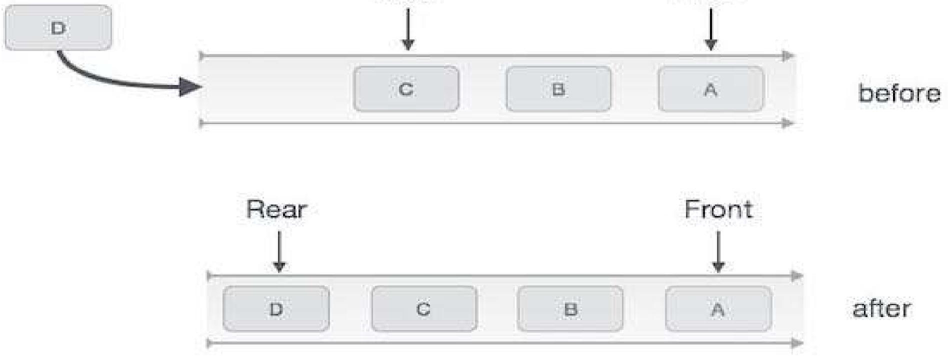
کردارەکان Operations کە لەگەڵ نۆبە گرتن Queue جێبەجێدەکرێت، ئەمانەی لای خوارەوەن کە بە گوێرەی پێویستیان شییان دەکەینەوە و باسیان دەکەین:

کرداری زیاد کردنی دانە Enqueue () Operation

ئەم کردارە بۆ زیاد کردن Add یان بلێین پاشەکەوت کردنی Store دانەکان بەکاردێت بۆناو نۆبە گرتنەکە Queue، کە دوو ئاماژە پێدەرمان هەیە بۆ ئەم مەستە، ئەویش دوواوە Rear ((پێشی دەلێن : کلیك Tail)) لەگەڵ پێشەوە Front ((پێشی دەلێن: سەر Head))، ئەم کردارەش زیاتر قورسە وەك کرداری زیاد کردن لە شارای ڕەفەیی Stackدا.

ئەم هەنگاوانەی خوارەوە پێویستن بۆ ئەنجامدانی کرداری خستنە ناوی Insert دانەیەك بۆ ناو نۆبەکە، کە پێی دەلێین: زیادکردنی دانە Enqueue.

1- پشکنین ئەگەر نۆبەکە پڕبێت Check If Queue is Full.
2- ئەگەر سەرەکە ((نۆبەکە)) پڕبێت، ئەوا سەر ڕووکەوتن Overflow ڕوو دەدات و دەردەچێت Exit لە نۆبەکە Queue.
3- ئەگەر نۆبەکە پڕ نەبێت، ئەوا ئاماژە پێدەری دوواوە Rear Pointer زیاد دەکات بۆ بۆشایی بەتاڵی داهاتوو Next Empty Space.
4- زانیاری دانەکە زیاد دەبێت بۆ شوێنی نۆبەکە Queue Location.
5- بەسەرکەوتووانە کارەکە بە ئەنجام دەکات .



Queue Enqueue

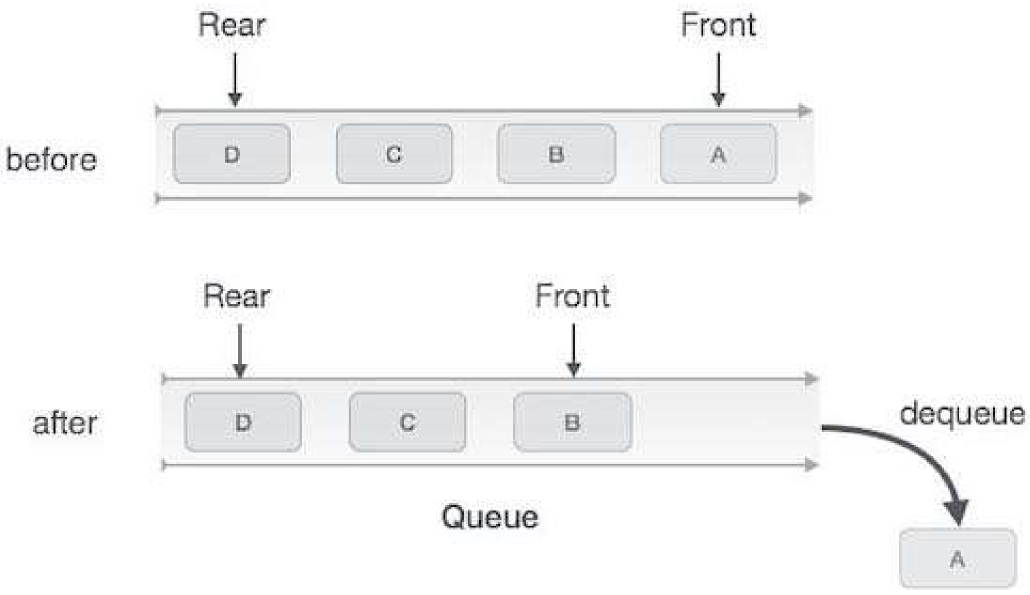
برووانە ئەلگۆریسمی کردارەکە:

```
procedure enqueue(data)
  if queue is full
    return overflow
  endif
  rear ← rear + 1
  queue[rear] ← data
  return true
end procedure
```

کرداری لابردنەی ناو نۆبەDequeue

ئەم کردارە بەکاردەهێنرێت بۆ دەستپێگەیشتنی Access دانەکانی ناو سەرەکە ((نۆبەکە)) و، لابردنی دانەکان Remove، بۆ جێبەجێکردنی کردارەکەش ئەم هەنگاوانە جیبەجێدەکەین:

1- پشکنین ئەگەر نۆبەکە پڕبێت Check If Queue is Full.
2- ئەگەر سەرەکە ((نۆبەکە)) پر نەبێت، ئەوا سەر ڕووکەوتن Overflow ڕوو نادات و دەردەچێت Exit لە نۆبەکە Queue.
3- ئەگەر نۆبەگرتنەکە بەتاڵ نەبێت، ئەوا دەست گەیشتن Access بە زانیارییەکە لە شوێنی پێشەوەدا Front کە ئاماژەی بۆ کرابێت.
4- ئاماژە پێدەی پێشەوە Front Pointer زیاد دەکات بۆ دانەی چالاکی دوواتر Next Data Element.
5- بەسەرکەوتووانە کارەکە رووی دا.



Queue Dequeue

بڕِوانە ئەلگۆریسمی کردارەکە:

```
procedure dequeue
  if queue is empty
    return underflow
  end if
  data = queue[front]
  front ← front - 1
  return true
end procedure
```

کرداری گەیشتنە لوتکە Peak () Operation

گەیشتنە پەری ئەو پەر ()Peak: ئەم کردارە بەکاردێت بۆ گەراندنەوەی نرخی لوتکە ((پۆپە)) Peak ی نۆبەکە ، واتە ئەو دانەیەی دەکەوێتە پێشەوەی Front سەرەکە Queue.

بڕوانە ئەلگۆریسمەکە:

```
begin procedure peek
  return queue[front]
end procedure
```

کرداری پڕە isFull() Operation

پشکنینی حەجمی نۆبەکە Queue بۆ پڕێتی ()isFull: نۆبە گرتنەکە دەپشکنێت ئەگەر پڕ بێت.
بڕوانە ئەلگۆریسمی کردارەکە:

```
begin procedure isfull
  if rear equals to MAXSIZE
    return true
  else
    return false
  endif
end procedure
```

کرداری بەتاڵە isEmpty() Operation

پشکنینی حەجمی نۆبەکە بۆ بەتاڵێتی isEmpty(): نۆبە گرتنەکە Queue دەپشکنێت ئەگەر بەتاڵ بێت. بڕوانە ئەلگۆریسمەکە:

```
begin procedure isempty
  if front is less than MIN  OR front is greater than rear
    return true
  else
    return false
  endif
end procedure
```

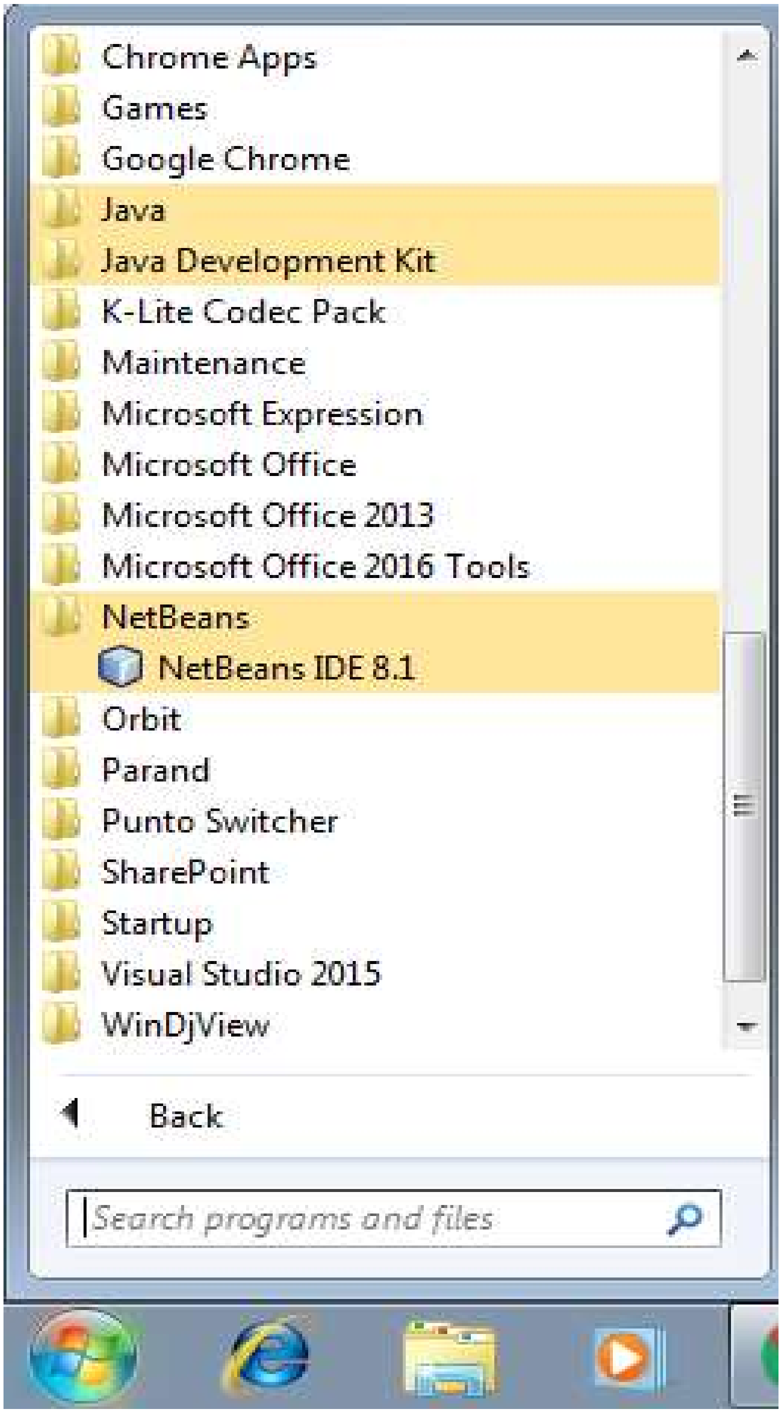
جێبەجێکردنی شارای نۆبە

Queue Implementation

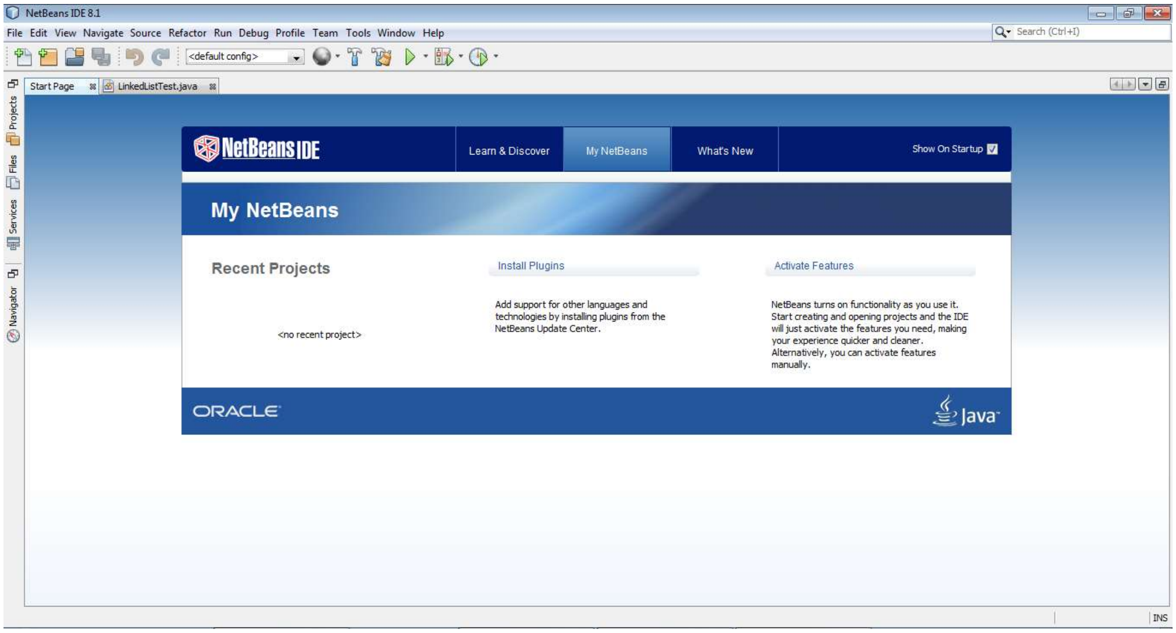
جێبەجێکردنی شارای نۆبەیی

Queue Implementation

1-بەرنامە نێتبین NetBeans دەکەینەوە، بە کلیك کردن لەسەر دووگمەی دەستپێکردن Start Button، پاشان کلیك لەسەر NetBeans و دوواتر کلیك لەسەر NetBeans IDE 8.1:

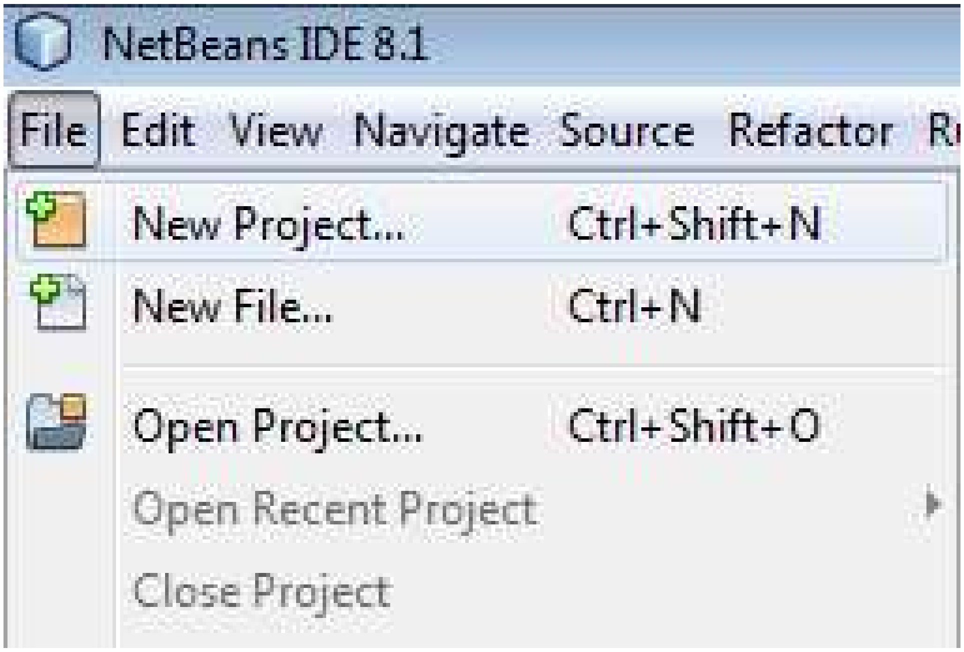


2-ڕووکاری بەرنامەی نێتبینز NetBeansدەکرێتەوە.

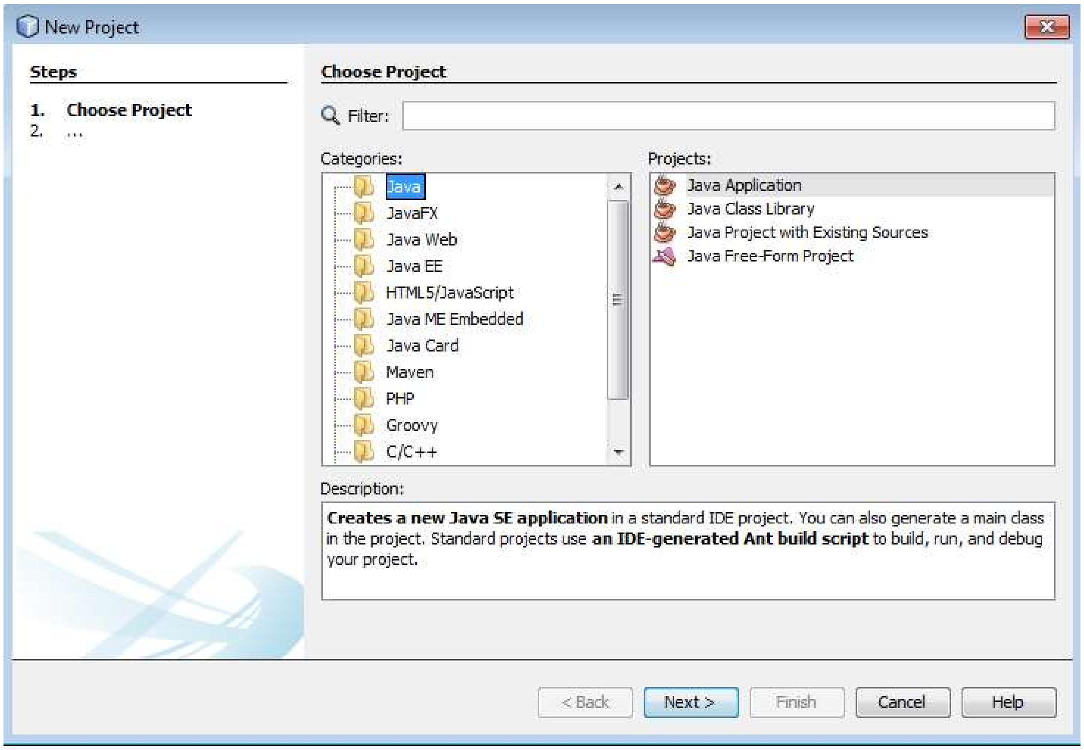


3-کلیك لەسەر دووگمەی فایل Fileدەکەین، بۆ ئەوەی پرۆژەیەکی نوێ درووست بکەین.

4-کلیك لەسەر پرۆژەی نوێ New Project دەکەین:



5-ڕووکاری تایبەت بە پرۆژەی نوێ New Project دەکرێتەوە و، کلیك لەسەر دووگمەی دوواتر Next دەکەین.



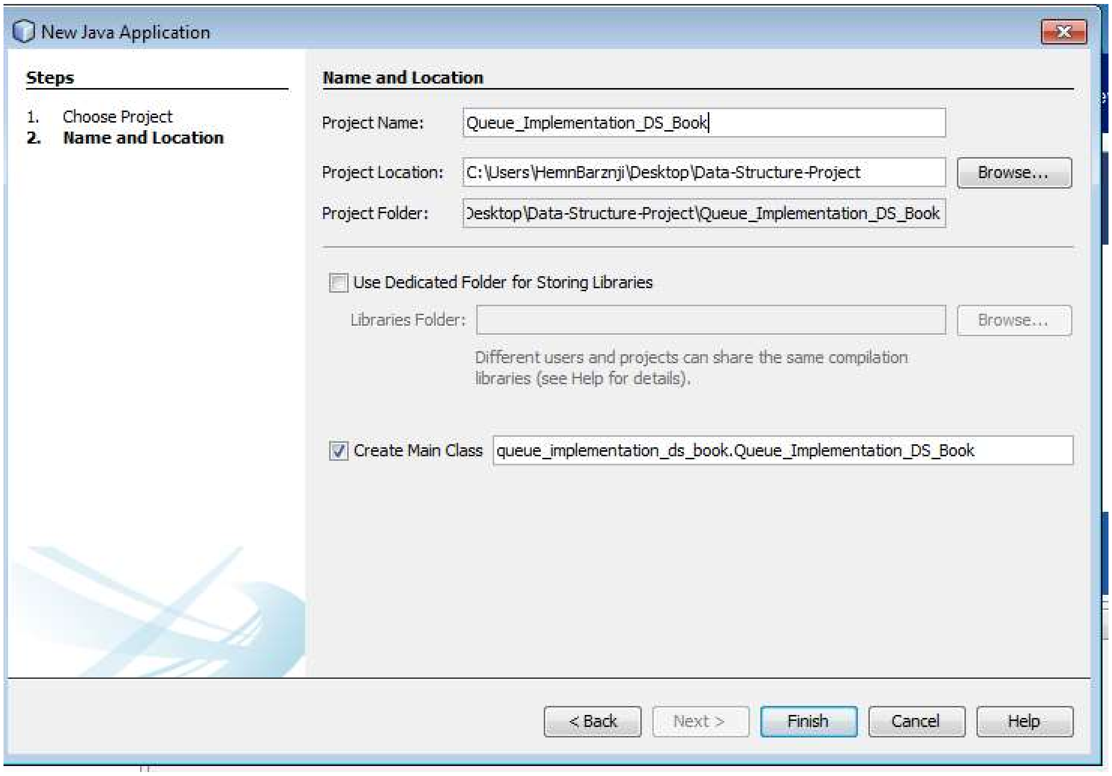
6-ڕووکارێکی نوێ دەکرێتەوە:

-لەبەشی ناوی پرۆژە Project Nameدا، ناوێك بۆ پرۆژەکە بنووسە.

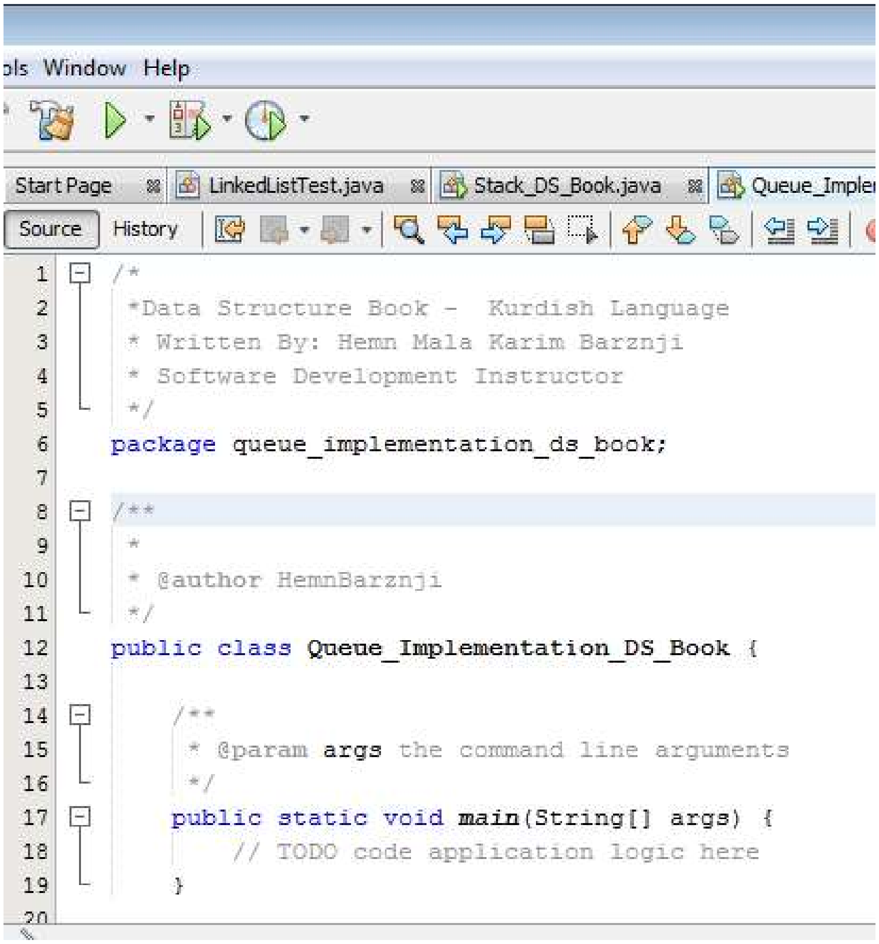
-لەبەشی شوێنی پرۆژە Project Locationدا، شوێنی پرۆژەکە دیاری بکە.

-کلیك لەناو سنووقی پشکنینی Check Box دروستکردنی پۆلی سەرەکی Create Main Class بکە، ئەگەر دەتەوێت پۆلێکت بۆ درووست بکات بە ناوێکەوە کە بەرنامەکە دیاری دەکات.

-لە دووای ئەو هەنگاوانە کلیك لەسەر دووگەی کۆتای هاتن Finishبکە.



7-ئیستا پرۆژەکە درووست بوو:



8-بەمەش پۆلێك Class درووست بوو، كە ڕێگەی سەرەكیشی Main Class تیایە.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
package queue_implementation_ds_book;

/**
 @author HemnBarznji
 */
public class Queue_Implementation_DS_Book {

    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

کۆد نووسی بۆ شارای نۆبە

Coding for Queue

وەك لە بابەتی شارای نۆبە گرتن Queueدا، باسمان کرد، دەتوانین ڕیزکراوە Array، یان لیستەی پێکەوە بەستراوو Linked List، یاخود شارای ڕەفەیی Stack، بەکاربهێنین بۆ درووستکردنی شارای نۆبە گرتن Queue.

1-سەرەتا پاکەتی جاڤا یوتیلْ Java Util دەهێنینە ناوەوە:

```
import  java.util.*;
```

2-ڕاگەیاندن و بڵاوکردنەوەی شارای نۆبە گرتن Queue Declaration ئەنجام دەدەین، لەناو ڕێگەیی سەرەکی Main Method دا، بەبەکارهێنانی لیستەی پێکەوە بەستراوو Linked List، بڕوانە کۆدەکەی خوارەوە:

```
    //Creating Queue
    Queue<String> queue = new LinkedList<String>();
```

شارای نۆبە گرتن Queue چاپ Print دەکەین کە بەتاڵە و هیچی تیا نییە. چونکە هیچ دانەیەکمان داغڵ نەکردووە بۆی.

```
    //Print Empty Queue
     System.out.println("Queue ="+ queue);
```

3-پاشان کرداری داغڵکردنی زانیاری ((نۆبە دەگریّ Enqueue)) ئەنجام دەدەین، بە هۆیی ڕێگەیی زیاد کردن لە دوواوەوە Rear ((لە کلکەوە Tail)) Offer Method دانەکان زیاد دەکەین بۆ ناو شارای نۆبەگرتن Queue، کە سەرەتا ناوی شارای نۆبە Queue دەنووسین و پاشان نووقتە دادەنێین و وشەی زیاد کردن offer. بە دووایدا دەنووسین و لە نێوان کەوانە و دووانە کۆمای سەرەوەدا (“   ”) ئەو نرخە دەنووسین کە دەمانەوێت داغڵی بکەین:

```
queue.offer("Hemn");
```

4-بەهەمان شێوە هەموو دانەکانی تر زیاد دەکەین کە دەمانەوێت:

```
queue.offer("Barznji");
queue.offer("May - 1 1982");
```

5-ڕێگەی پیشاندان Display Method دەنووسین، بۆ چاپکردن:

```
System.out.println("Queue ="+ queue);
```

6-ڕێگەی حەجم size() بەکاردەهێنین بۆ زانینی حەجمی نۆبەکە Queue:

```
System.out.println("Size: " + queue.size());
```

7.بەکارهێنانی هەردوو ڕێگەی peak() و element() بۆ گەڕاندنەوەی نرخی سەرە head ی شارای نۆبە گرتن Queue، بەڵام لای نابات Remove.

```
System.out.println("Queue head using peek   : " + queue.peek());
System.out.println("Queue head using element: " +
queue.element());
```

8-پاشان کرداری لە نۆبە دەرچوون Dequeue ئەنجام دەدەین، بە هۆی ڕێگەی گەڕاندنەوە و لابردنی دانەکان لە سەرەوە ((Head)) واتە لە پێشەوە ((Front))، ئەم ڕێگەیە بەکاردەهێنین بۆ لابردنی دانەکان لە پێشەوە، بەمەش یەکەم دانەی داغڵبوو First – In، یەکەم دانە دەبێت لە نۆبەکە دێتە دەر First – Out:

```
queue.poll();
```

بە ژمارەی دانەکانی ناو نۆبەکە، ڕێگەی دەرچوون لە نۆبەکە بەکاردەهێنین یان لە جیاتی ئەمە، لە ڕێگەی سورەکانەوە Loop هەمان کار دەکەین بۆ ئەوەی بە ژمارەی دانەکانی ناو نۆبەکە، ڕێگەی دەرچوون لە نۆبەکە نەنووسین:

```
Object data;
```

```
    while ((data = queue.poll()) != null) {
      System.out.println(data);
    }
```

تێبینی::

دەتوانین ڕێگەی لابردن remove() بە کاربهێنین بۆ لابردنەوەی دانەکان لە ناو نۆبەکە، لە جیاتی ڕێگەی poll() دا.

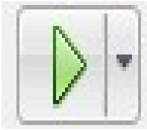
ئەمەش هەموو کۆدەکەیە پێکەوە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
package queue_implementation_ds_book;

/**
 @author HemnBarznji
 */
import java.util.*;
public class Queue_Implementation_DS_Book {

    public static void main(String[] args) {
```

```
//Creating Queue
Queue<String> queue = new LinkedList<String>();
//Print Empty Queue
System.out.println("Queue ="+ queue);

queue.offer("Hemn");
queue.offer("Barznji");
queue.offer("May - 1 1982");
System.out.println("Size: " + queue.size());
 System.out.println("Queue ="+ queue);

System.out.println("Queue head using peek   : " + queue.peek());
System.out.println("Queue head using element: " +
queue.element());

 queue.poll();
System.out.println("Queue After First  Poll="+ queue);
 queue.poll();
System.out.println("Queue After Second  Poll="+ queue);

queue.poll();
System.out.println("Queue After  Third Poll="+ queue);
```

```
/*
     Object data;
   while ((data = queue.poll()) != null) {
     System.out.println(data);
   }
   */
   }
}
```

9-کلیك لەسەر دووگمەی جێبەجێکردنی پرۆژە Run Projectدەکەین:



10-بەرهەم و ئەنجام::

```
Queue =[]
Size: 3
Queue =[Hemn, Barznji, May - 1 1982]
Queue head using peek   : Hemn
Queue head using element: Hemn
Queue After First  Poll=[Barznji, May - 1 1982]
Queue After Second  Poll=[May - 1 1982]
Queue After  Third Poll=[]
```

گۆرینی نۆبە بۆ لیست

Convert Queue to List

بە بەکارهێنانی پۆلی لیست بە وەرگرتنی تەنێك لێی Object و پاشان بەکارهێنانی لووپێك کارەکە ئەنجام دەدەین و نۆبەکە Queue دەکەین بە لیست:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
package queue_implementation_ds_book;


/**
 @author HemnBarznji
 */
import java.util.*;
public class Queue_Implementation_DS_Book {


   public static void main(String[] args) {
      //Creating Queue
      Queue<String> queue = new LinkedList<String>();
      //Print Empty Queue
```

```
        System.out.println("Queue ="+ queue);


        queue.offer("Hemn");
        queue.offer("Barznji");
        queue.offer("May - 1 1982");
        System.out.println("Size: " + queue.size());
         System.out.println("Queue ="+ queue);


    List<String> myList = new ArrayList<String>(queue);


    for (Object queueList : myList)
      System.out.println(queueList);
    }
}
```

دووای کلیك کردن لەسەر دووگمەی جێبەجێکردنی پرۆژە Run Project ئەنجامەکە دەبینین:

```
Queue =[]
Size: 3
Queue =[Hemn, Barznji, May - 1 1982]
Hemn
Barznji
May - 1 1982
```

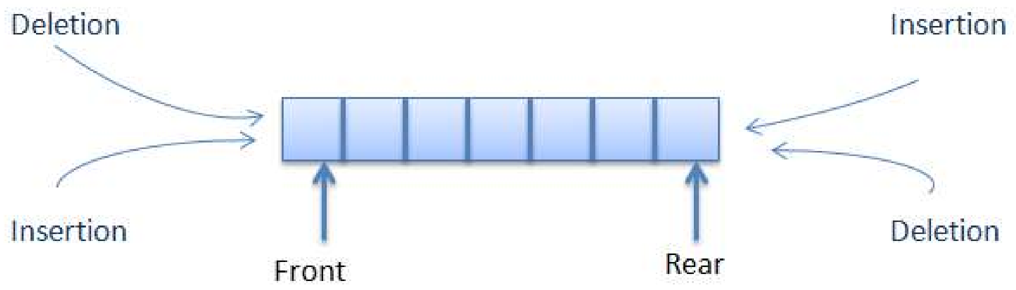
شارای

نۆبە بەستن لە هەردوو لاوە

Double – Ended Queue

((Deque))

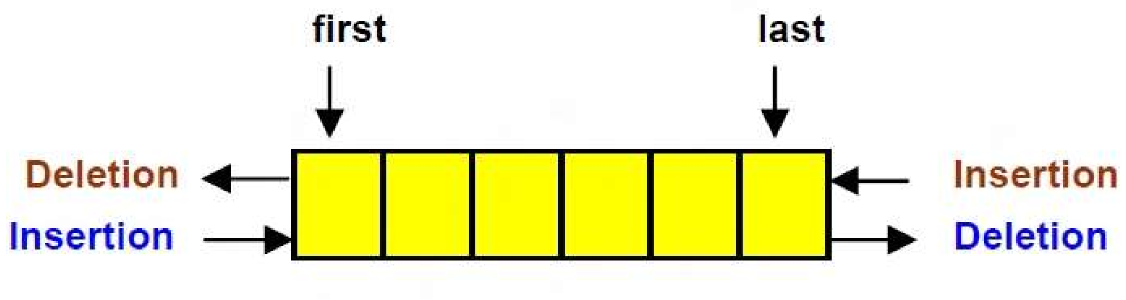
شارای نۆبە بەستن لە هەردوو لاوە Double – Ended Queue کە پێشی دەوترێت Deque، ئەو شارایەیە کە پاڵپشتی داغڵکردن Insertion و سرینەوە Deletion ی دانە پێکهێنەرەکان Item دەکات لە هەردوو لاوە، واتە لە پێشەوە Front و لە دوواوە Rear دانەی بۆ داغڵ دەکرێت و دانەی لێدەسرێتەوە.



دروستکردنی دیکیوو

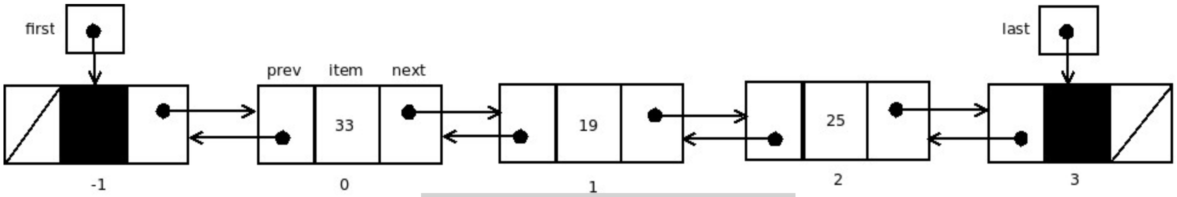
Building Deque

دەتوانین دیکیوو دروست بکەین، بە بەکارهێنانی ریزکراوە Array، کە لە زۆربەی بەرنامە سازییەکان دا، پۆلی تایبەت بە دروستکردنی دیوو بە ریزکراوە هەیە، بۆ نموونە لە زمانی بەرنامە سازیی جاڤادا، پۆلی ArrayDeque دابین کراوە بۆ ئەم مەبەستە و ، هەموو رێگەکانمان Methods بۆ ئامادەکراوە و، دەتوانین بەکاری بهێنین.



هەروەها دەتوانین لیستەی پێکەوەبەستراووی دوو ئاراستە Doubly Linked List بەکاربهێنین بۆ دروست کردنی دیکیوو Deque، بە ناساندنی گرێی سەرە Head Node ((گرێی پێشەوە

Front Node)) و گرێی کلك ((Tail Node ((گرێی دوواوە Rear Node))، پاشان زیاد کردن یان سرینەوە لە هەردوو لاوە، بە گوێرەی پێویست کە ڕێگە پێدراوە لە هەردوو لاوە و هەردوو لا کراوەیە بۆ سەرە گرتن،



بەکارهێنانەکانی دیکیوو

Deque Applications

1- بەکاردێت بۆ دروستکردنی شارای ڕەقەیی Stack. واتە وەکو شارای ڕەفەیی Stack بەکاردێت.

2- بەکاردێت بۆ دروستکردنی شارای نۆبە بەستن Queue. واتە وەکو شارای نۆبە بەستن Queue بەکاردێت.

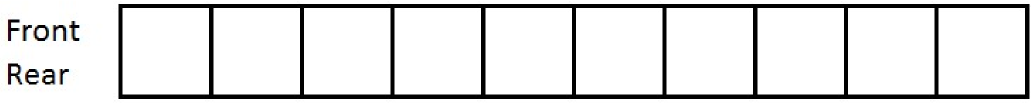
3- کردارەکانی Undo و Redo لە بەرنامەکان دا.

4- بۆ تاقیکردنەوە و هەڵسەنگاندنی کەش و هەوا بەکاردێت.

5- گەران و تاقیکردنەوەی دوو سەر ((دوو لا)).

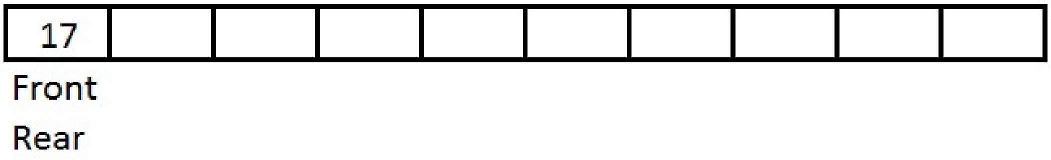
Example نموونە

دیکیووییەك دەناسێنین کە لە 10 خانە پێمهاتبێت و، لە سەرەتادا بەتاڵە و هیچ زانیارییەکی تیانییە بۆیە پێشەوە Front یەکسانە بە دوواوە Rear و ئەمیش یەکسانە بە -1:

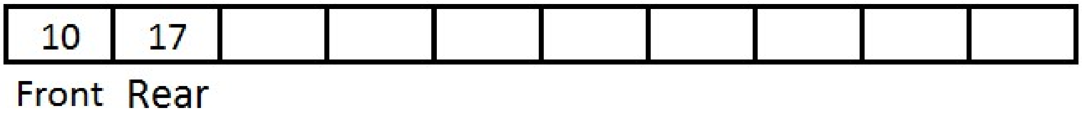


Front=Rear=-1
Deque is Empty

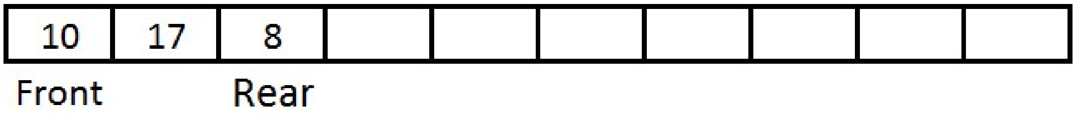
ئەگەر ژمارە 17 داغڵبکەین، ئەوا پێشەوە و دوواوە Front – Rear، ئاماژە دەکەن بە ژمارە حەڤدەکە:



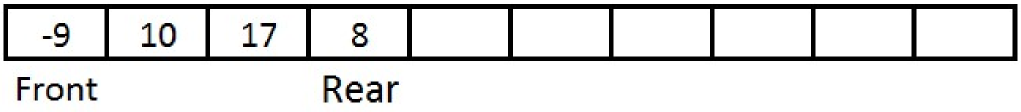
ئەگەر ژمارە 10 لە پێش ژمارە 17 وە داغڵ بکەین ، ئەوا ژمارە 10 دەبێتە پێشەوە Front و ژمارە 17 دەبێتە دوواوە Rear:



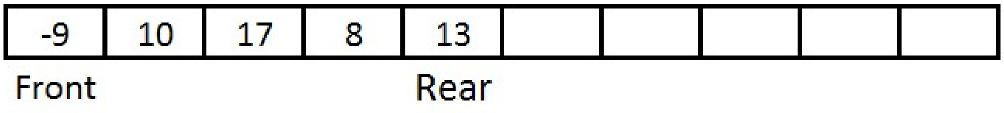
ئەگەر ژمارە 8 زیاد بکەین لە دووای ژمارە 17، ئەوا دوواوە Rear ئاماژە بە 8 دەکات و دوواوە دەبێتە ژمارە هەشت:



ئەگەر 9- لە سەرەتاوە زیاد بکەین ئەوا هەموو ژمارەکانی تر بەرەو لای راست دەچولێن و شوینی پێشەوە و دوواوەش دەگۆرێت:



پاشان ژمارە 13 لە پشتەوە داغڵدەکەین و بەمەش دوواوە Rear ئاژە بە ژمارە 13 دەکات:



ئەگەر ژمارە 28 بۆ لە پێشەوە زیاد بکەین، ئەوا هەموو دانەکانی تر بەرەو لای راست دەجولێن و پێشەوە Front دەبێتە ژمارە 28:

28	-9	10	17	8	13				
Front					Rear				

پاشان ژمارە 14 لە دوواوە زیاد دەکەین و، بەمەش ژمارە 14 دەبێتە پشتەوە Rear:

28	-9	10	17	8	13	14			
Front						Rear			

ئەگەر ژمارە 5 لە سەرەتاوە زیاد بکەین، ئەوا هەموو ژمارەکانی تر بەلای راست دا، دەجولێن و، پێشەوە Front ئاماژە بە ژمارە پێنجەکە دەکات، و دوواوە Rearدەگوێزریتەوە بە پێی جولانەکە:

5	28	-9	10	17	8	13	14		
Front							Rear		

ئێستا ئەگەر ژمارە 25 لە کۆتاییەوە زیاد بکەین، ئەوا دوواوە Rear ئاماژە بە ژمارە بیست و پێنج دەکات :

5	28	-9	10	17	8	13	14	25	
Front								Rear	

ئەگەر ژمارە 6 لە سەرەتاوە زیاد بکەین، ئەوا هەموو دانەکان بەلای راست دا، دەجولێن و، دوواوە Rear ئاماژە بە دووا ژمارە دەکات و پێشەوە Front ئاماژە بەیەکەم ژمارە دەکات:

6	5	28	-9	10	17	8	13	14	25
Front									Rear

ئیستا ئەگەر بمانەوێت ژمارە 21 لە کۆتاییەوە زیاد بکەین، پێمان دەڵێت دیکیووەکە پڕە. چونکە تەنها 10 خانەمان هەبوو، ئەو 10 خانەیەش پربوون بە دە ژمارە:

6	5	28	-9	10	17	8	13	14	25
Front									Rear

IF Add 21 in Rear
Deque is Full

بەهەمان شێوە ئەگەر بمانەوێت ژمارە 11 زیاد بکەین، لە پێشەوە ئەوا پێمان دەڵێت پڕە و زیادی ناکات:

6	5	28	-9	10	17	8	13	14	25
Front									Rear

IF Add 11 in Front
Deque is Full

بەهەمان شێوەی زیاد کردنی دانەکان، دەتوانین بیان سرینەوە، هەم لە پێشەوە یان لە دواوە، یان لە هەردوو لاوە، بە گوێرەی پێویستی کارەکە:

6	5	28	-9	10	17	8	13	14	25
Front									Rear
	5	28	-9	10	17	8	13	14	25
	Front								Rear
	5	28	-9	10	17	8	13	14	
	Front							Rear	
		28	-9	10	17	8	13	14	
		Front						Rear	
		28	-9	10	17	8	13		
		Front					Rear		

جێبەجێکردنی

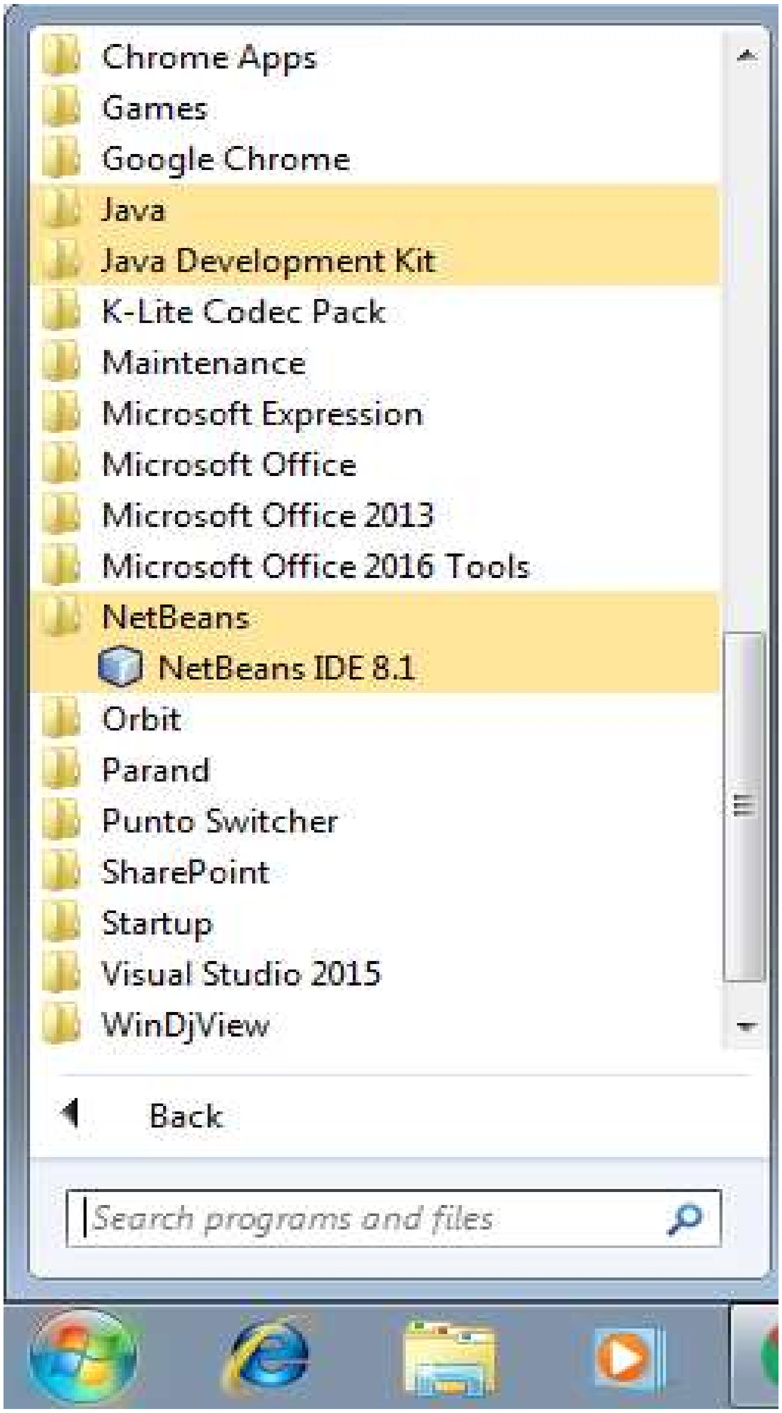
دیکیو

Deque Implementation

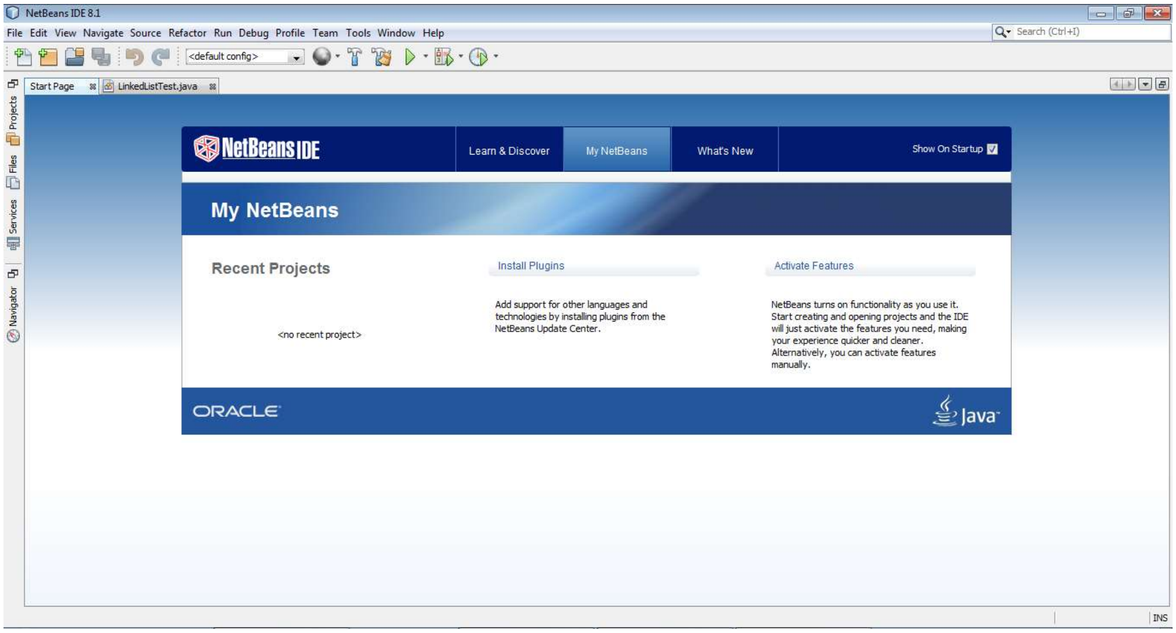
جێبەجێکردنی شارای نۆبەیی لە هەردوو لاوە

Deque Implementation

1-بەرنامە نێتبین NetBeans دەکەینەوە، بە کلیك کردن لەسەر دووگمەی دەستپێکردن Start Button، پاشان کلیك لەسەر NetBeans و دوواتر کلیك لەسەر NetBeans IDE 8.1:

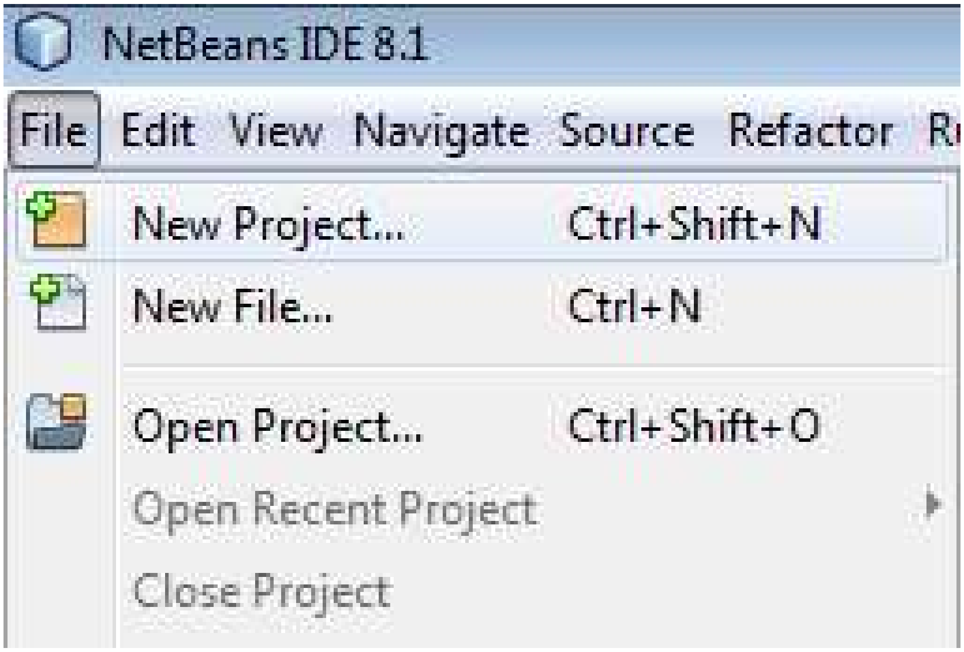


2-ڕووکاری بەرنامەی نێتبینز NetBeansدەکرێتەوە.

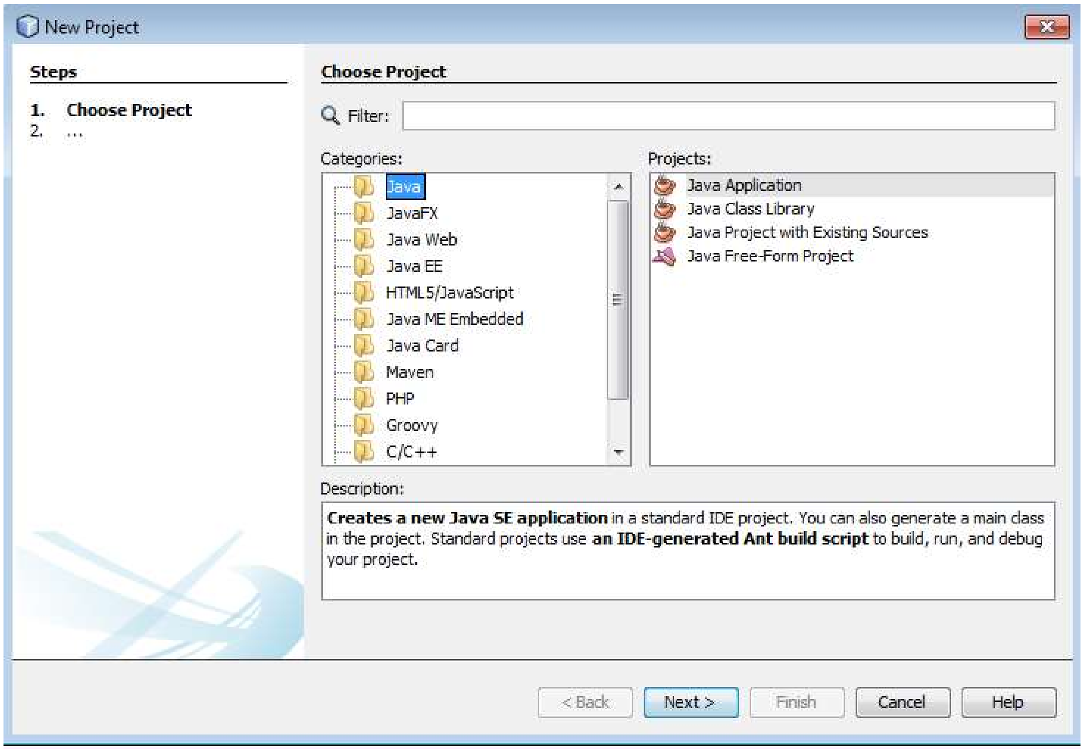


3-کلیك لەسەر دووگمەی فایل Fileدەکەین، بۆ ئەوەی پرۆژەیەکی نوێ درووست بکەین.

4-کلیك لەسەر پرۆژەی نوێ New Project دەکەین:



5-ڕووکاری تایبەت بە پرۆژەی نوێ New Projectدەکرێتەوە و، کلیك لەسەر دووگمەی دوواتر Next دەکەین.



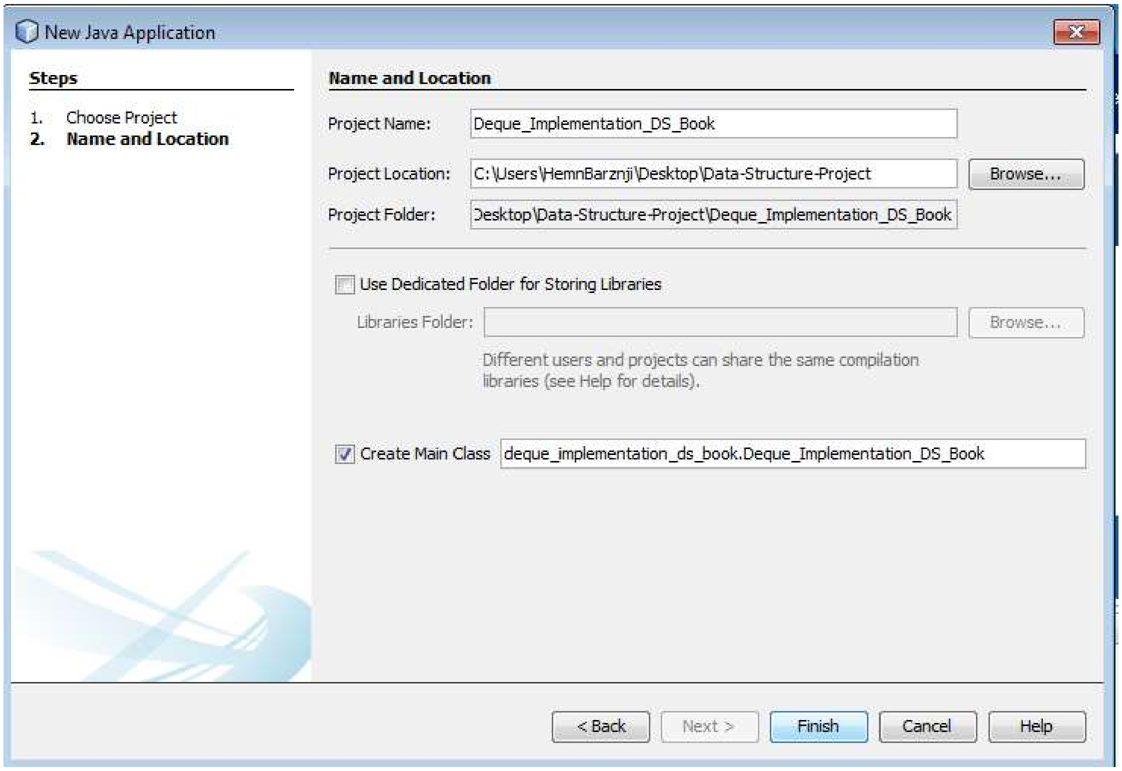
6-ڕووکارێکی نوێ دەکرێتەوە:

-لەبەشی ناوی پرۆژە Project Nameدا، ناوێك بۆ پرۆژەکە بنووسە.

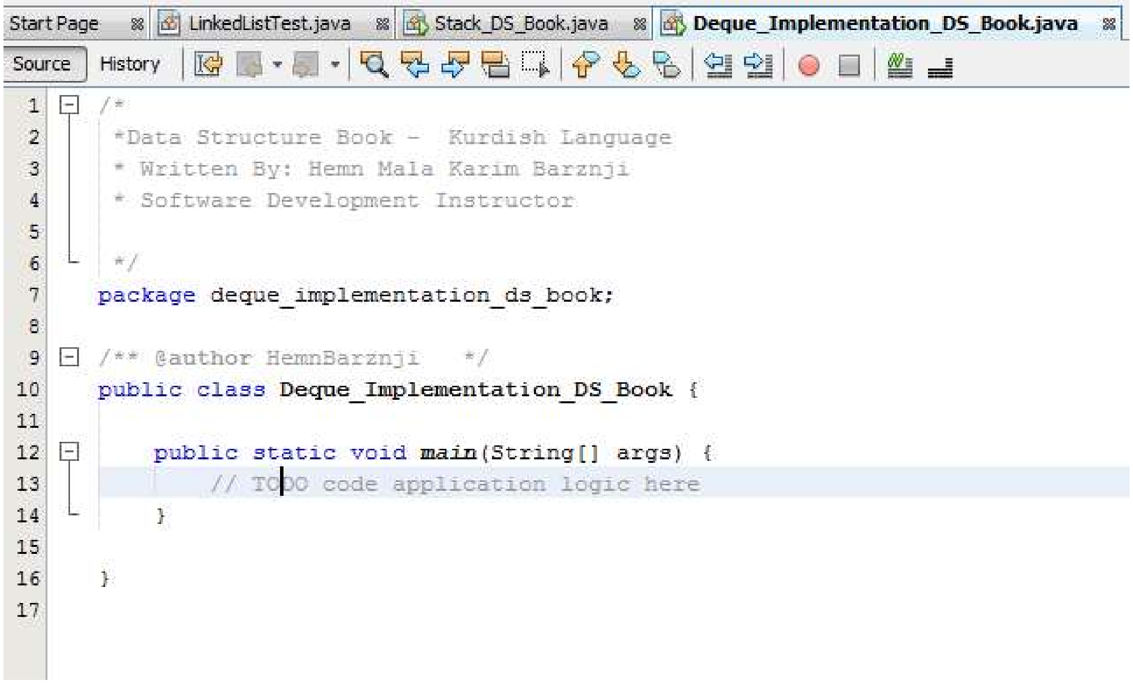
-لەبەشی شوێنی پرۆژە Project Locationدا، شوێنی پرۆژەکە دیاری بکە.

-کلیك لەناو سنووقی پشکنینی Check Box درووستکردنی پۆلی سەرەکی Create Main Class بکە، ئەگەر دەتەوێت پۆلێکت بۆ درووست بکات بە ناوێکەوە کە بەرنامەکە دیاری دەکات.

-لە دووای ئەو هەنگاوانە کلیك لەسەر دووگەی کۆتای هاتن Finishبکە.



7-ئیستا پرۆژەکە درووست بوو:



8-بەمەش پۆلێك Class دروست بوو، كە رێگەی سەرەكیشی Main Class تیایە.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
package deque_implementation_ds_book;

/** @author HemnBarznji   */
public class Deque_Implementation_DS_Book {

    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

کۆد نووسی بۆ شارای نۆبەیی لە هەردولاوە

Coding for Deque

وەك لە پێشتردا، باسمان کرد، دەتوانین ڕیزکراوە Array، یان لیستەی پێکەوە بەستراوو Linked List، بەکاربهێنین بۆ دروستکردنی شارای نۆبە گرتن لە هەردوو لاوە Deque، بەڵام ئێمە ئەو پۆلە Class ئامادەیە بەکاردەهێنین کە لە پاکەتی جاڤا یوتیلّ دا هەیە.

1-سەرەتا پاکەتی جاڤا یوتیلّ Java Util دەهێنینە ناوەوە:

```
import  java.util.*;
```

2-ڕاگەیاندن و بڵاوکردنەوەی شارای نۆبە گرتن لە هەردوو لاوە Deque Declaration ئەنجام دەدەین، لەناو ڕێگەیی سەرەکی Main Method دا، بۆ مەبەستی دروستکردنی دیکیووی بەتاڵ، بڕوانە کۆدەکەی خوارەوە:

```
// create an empty array deque with an initial capacity
Deque<Integer> deque = new ArrayDeque<Integer>(5);
```

شارای نۆبە گرتن لە هەردوو لاوە Deque چاپ Print دەکەین کە بەتاڵە و هیچی تیا نییە. چونکە هیچ دانەیەکمان داغڵ نەکردووە بۆی.

```
//Print Empty Queue
 System.out.println("Deque ="+ deque);
```

3-پاشان کرداری داغڵکردنی زانیاری ئەنجام دەدەین، بە هۆیی ڕێگەیی زیاد کردنەوە:

```
deque.add(20);
```

4-بەهەمان شێوە هەموو دانەکانی تر زیاد دەکەین کە دەمانەوێت:

```
deque.add(30);
deque.add(20);
```

```
deque.add(30);
deque.add(15);
deque.add(22);
deque.add(11);
```

5-ڕێگەی پیشاندان Display Method دەنووسین، بۆ چاپکردن:

```
//Print After  Adding
 System.out.println("Deque ="+ deque);
```

هەروەها دەتوانین لە ڕێگەی بەکارهێنانی سوڕەکانەوە Loops کارەکە ئەنجام بدەین و، دانەکان چاپ بکەین:

```
// let us print all the elements available in deque
 for (Integer number : deque) {
 System.out.println("Number = " + number);
 }
```

6-ڕێگەی حەجم ()size بەکاردەهێنین بۆ زانینی حەجمی نۆبەکە Queue:

```
System.out.println("Size: " + deque.size());
```

7-دەتوانین ڕێگەی لابردن ()remove بە کاربهێنین بۆ لابردنەوەی دانەکان لە ناو شارای نۆبە لە هەردوو لاوە:

```
// Remove Item
 deque.remove();
```

ئەمەش هەموو کۆدەکەیە پێکەوە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */

package deque_implementation_ds_book;

import  java.util.*;

/** @author Hemn Barznji   */
public class Deque_Implementation_DS_Book {

    public static void main(String[] args) {
      // create an empty array deque with an initial capacity
   Deque<Integer> deque = new ArrayDeque<Integer>(5);

          //Print Empty Queue
        System.out.println("Deque ="+ deque);
```

```
  // use add() method to add elements in the deque
deque.add(20);
deque.add(30);
deque.add(20);
deque.add(30);
deque.add(15);
deque.add(22);
deque.add(11);

   //Print After  Adding
     System.out.println("Deque ="+ deque);

 System.out.println("Size: " + deque.size());

   System.out.println("Print Deque With For Loop");
 // let us print all the elements available in deque
for (Integer number : deque) {
System.out.println("Number = " + number);
}
```

```
  // Remove Item
  deque.remove();
   //Print After  Remove
       System.out.println("Deque After Remove ="+ deque);
   }
}
```

8-كليك لەسەر دووگمەی جێبەجێکردنی پرۆژە Run Projectدەکەین:

9-بەرهەم و ئەنجام::

```
Deque =[]
Deque =[20, 30, 20, 30, 15, 22, 11]
Size: 7
Print Deque With For Loop
Number = 20
Number = 30
Number = 20
Number = 30
Number = 15
Number = 22
Number = 11
Deque After Remove =[30, 20, 30, 15, 22, 11]
```

دروستکەرەکانی پۆل

Class Constructors

پۆلی شارای نۆبەیی لە هەردوو لاوە کە بە ریزکراوە دروست کراوە و ناوی Array Deque، چەند دروستکەرێکی Constructor هەیە، کە ئەمانەی لای خوارەوەن:

- دروستکەری ArrayDeque()

ئەم دروستکەرە Constructor بەکاردێت بۆ دروستکردنی دیکیویی بە ئەرەیی دروستکراوی بەتاڵ Empty Array Deque، بە توانایەکی سەرەتایی تەواو بۆ 16 دانەی پێکهێنەر.

- دروستکەری ArrayDeque(Collection<?extends E>c)

ئەم دروستکەرە Constructor بەکاردێت بۆ دروستکردنی دیکیویی بە ئەرەیی دروستکراو Array Deque، کە دانە پێکهێنەرەکانی کۆکراوەیەکی دیاری کراوی Specified Collection تیابێت.

- دروستکەری ArrayDeque(int numElements)

ئەم دروستکەرە Constructor بەکاردێت بۆ دروستکردنی دیکیویی بە ئەرەیی دروستکراوی بەتاڵ Empty Array Deque، لەگەڵ دیاری کردنی توانای سەرەتایی بۆ هەڵگرتنی دانە پێکهێنەرەکان.

ڕێگەکان Methods

بەهەمان شێوەی دروستکەرەکان Constructor، ڕێگەکانیش Methods دەبێت بناسێنرێن و باس بکرێن بۆ ئەوەی بە گوێرەی پێویست کەڵکیان لێوەربگیرێت و بەکاربهێنرێن. ژمارەی ڕێگە ئامادەکراوەکان 31 دانەیە، کە بە ئاسانی دەتوانین بەکاریان بهێنین بۆ ئەنجامدانی کارەکانمان.

هەموو ڕێگەکان لەسەر نموونەکەی سەرەوە جیبەجێدەکەین بۆ ئەوەی کەمترین کۆد بنووسینەوە، بەلام لەهەر شوێنێکدا بە پێویستمزانی هەموو کۆدەکە دەنووسمەوە.

ڕێگەی زیادکردنی دانەپێکهێنەر لە جۆریی بولییەن

Boolean add(E e);

ئەم ڕێگەیە Method بەکاردێت بۆ سەربارکردن و زیادکردنی دانەیەکی پێکهێنەر بۆ کۆتایی End دیکیوو کە نرخی ڕاست True دەگێڕێتەوە بۆمان. ئەگەر دانەیەکی دیاری کراوو بەتاڵ Null بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException پێویستە.

نموونە:

```
// use add() method to add elements in the deque
  deque.add(20);
  deque.add(30);
  deque.add(20);
  deque.add(30);
  deque.add(15);
  deque.add(22);
  deque.add(11);
```

ئەنجام//

```
Deque =[]
Deque =[20, 30, 20, 30, 15, 22, 11]
```

زیادکردن لە سەرەتاوە

Void addFirst(E e);

ئەم ڕێگەیە بەکاردەهێنرێت بۆ دانانی Insert دانەیەکی پێکهێنەری دیاری کراوو لە پێشەوەی Front دیکیوو، ئی e ئەو دانە پێکهێنەرەیە کە زیاد دەکرێت. ئەگەر دانەیەکی دیاری کراوو بەتاڵ Null بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException پێویستە.

```
// use addFirst() method to add element at the front of the deque
deque.addFirst(10);
deque.addFirst(15);
deque.addFirst(20);//now, element 20 will be at the front
```

ئەنجام//

Deque =[]

Deque =[20, 15, 10, 20, 30, 20, 30, 15, 22, 11]

زیادکردن لە کۆتاییەوە

Void addLast(E e);

ئەم ڕێگەیە بەکاردەهێنرێت بۆ دانانی Insert دانەیەکی پێکهێنەری دیاری کراوو لە کۆتایی Rear دیکیووەوە، ئی e ئەو دانە پێکهێنەرەیە کە زیاد دەکرێت. . ئەگەر دانەیەکی دیاری کراوو بەتاڵ Null بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException پێویستە.

```
// the values will be printed in the same order
deque.addLast(10);
deque.addLast(12);
```

ئەنجام:

Deque =[]

Deque Befor =[20, 30, 20, 30, 15, 22, 11]

Deque After =[20, 30, 20, 30, 15, 22, 11, 10, 12]

سڕینەوە

Void clear();

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لابردنی Remove هەموو دانەکانی دیکیوو. ئەم ڕێگەیە هیچ نرخێك ناگێڕێتەوە.

```
// finding size of this deque
  int retval = deque.size();
  System.out.println("Deque consists of "+ retval +" elements");


  System.out.println("Performing clear operation !!");
  deque.clear();
  retval = deque.size();
  System.out.println("Now, deque consists of "+ retval +" elements");
```

ئەنجام//

Deque =[]

Deque consists of 7 elements

Performing clear operation !!

Now, deque consists of 0 elements

لەبەرگرتنەوەی دیکیوو

Object clone

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لەبەرگرتنەوەی دیکیوویی دیاری کراو، بە گوێرەی پێویست، هیچ داغڵکەرێکی Parameter نییە، و حاڵەتی نائاسایی Exception ڕوونادات.

```
Deque<Integer> deque2 = deque.clone();
```

بڕوانە هەموو کۆدەکە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */

package deque_implementation_ds_book;

import  java.util.*;
import  java.util.Deque;
/** @author Hemn Barznji   */

public class Deque_Implementation_DS_Book {
```

```
   public static void main(String[] args) {

// Create empty array deque with an initial capacity
          ArrayDeque<Integer> deque = new ArrayDeque<Integer>(8);

     // Use add() method to add elements
     deque.add(100);
     deque.add(200);
     deque.add(300);
     deque.add(400);
     deque.add(500);

     Deque<Integer> deque2 = deque.clone();

     // Print Deque2 elements
     for (Integer num : deque2) {
        System.out.println("Element is " + num);
     }
   }
}
```

ئەنجام//

```
Element is 100

Element is 200

Element is 300

Element is 400

Element is 500
```

تیابوون

Boolean contain (object O)

ئەم ڕێگەیە نرخی ڕاست True دەگێڕێتەوە، لە حاڵەتی هەبوونی دانەیەکی دیاری کراوویی پێکهێنەر لە دیکیوو دا، داغڵکەری ئۆ O ئەو دانە پێکهێنەرەیە کە تاقیدەکرێتەوە و، بۆ ئەوەی بزانین هەیە؟ یان نییە لە لیستەکەدا. ئەگەر هەبێت True دەگێڕێتەوە، ئەگەر نەبێت False دەگێڕێتەوە.

```
// deque contains element 10
   boolean retval = deque.contains(10);
```

بڕوانە هەموو بەرنامەکە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */

package deque_implementation_ds_book;
```

```
import  java.util.*;

import  java.util.Deque;

/** @author Hemn Barznji   */

public class Deque_Implementation_DS_Book {


    public static void main(String[] args) {

      // create an empty array deque with an initial capacity

   Deque<Integer> deque = new ArrayDeque<Integer>();


        //Print Empty Queue

       System.out.println("Deque ="+ deque);




     // use add() method to add elements in the deque

   deque.add(20);

   deque.add(30);

   deque.add(20);

   deque.add(30);

   deque.add(15);

   deque.add(25);
```

```
deque.add(11);

//Print Deque
 System.out.println("Deque ="+ deque);

// deque contains element 10
boolean retval = deque.contains(10);

if (retval == true) {
System.out.println("element 10 is contained in the deque");
}
else {
System.out.println("element 10 is not contained in the deque");
}

// deque does not contain element 25
boolean retval2 = deque.contains(25);

if (retval2 == true) {
System.out.println("element 25 is contained in the deque");
}
```

```
  else {
  System.out.println("element 25 is not contained in the deque");
  }
   }
}
```

ئەنجام//

```
Deque =[]
Deque =[20, 30, 20, 30, 15, 25, 11]
element 10 is not contained in the deque
element 25 is contained in the deque
```

ڕێگەیی دووبارە کردنەوەی پێچەوانەیی

()Iterator<E> descendingIterator

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی دانە پێکهێنەرەکانی دیکیوو Dque لە ڕێکخستنێکی پێچەوانەیی دا، واتە دانەکان لە کۆتاییەوە بۆ سەرەتا ڕێکدەخات.

```
// set Iterator as descending
  Iterator x = deque.descendingIterator();
  // print list with descending order
  while (x.hasNext()) {
  System.out.println(x.next());
  }
```

یاخود ئەم شێوازە کۆد نووسییە بەکاردەهێنین:

```
//iterator() is used to print all the elements
  //next() returns the next element on each iteration
  System.out.println("printing elements using iterator:");

  for(Iterator itr = deque.iterator();itr.hasNext();)  {
  System.out.println(itr.next());
  }

  //descendingIterator() is used to print the elements in reverse order
  //next() returns the next element on each iteration
  System.out.println("printing elements in reverse order:");

  for(Iterator descItr = deque.descendingIterator();descItr.hasNext();) {
  System.out.println(descItr.next());
  }
```

بڕووانە هەموو کۆدەکە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
```

```
 * Software Development Instructor
 */

package deque_implementation_ds_book;

import  java.util.*;
import  java.util.Deque;
/** @author Hemn Barznji   */
public class Deque_Implementation_DS_Book {
   public static void main(String[] args) {
      // create an empty array deque with an initial capacity
   Deque<Integer> deque = new ArrayDeque<Integer>();

         //Print Empty Queue
        System.out.println("Deque ="+ deque);
     // use add() method to add elements in the deque
   deque.add(20);
   deque.add(30);
   deque.add(20);
   deque.add(30);
   deque.add(15);
```

```
deque.add(25);
deque.add(11);

 System.out.println(deque);
// set Iterator as descending
Iterator x = deque.descendingIterator();
// print list with descending order
while (x.hasNext()) {
System.out.println(x.next());
}

//iterator() is used to print all the elements
//next() returns the next element on each iteration
System.out.println("printing elements using iterator:");

for(Iterator itr = deque.iterator();itr.hasNext();)  {
System.out.println(itr.next());
}
//descendingIterator() is used to print the elements in reverse order
//next() returns the next element on each iteration
System.out.println("printing elements in reverse order:");
```

```
  for(Iterator descItr = deque.descendingIterator();descItr.hasNext();)
{
  System.out.println(descItr.next());
  }


   }
}
```

ئەنجام//

```
Deque =[]
[20, 30, 20, 30, 15, 25, 11]
11
25
15
30
20
30
20
printing elements using iterator:
20
30
20
```

30

15

25

11

printing elements in reverse order:

11

25

15

30

20

30

20

گەراندنەوەی سەرەی دیکیوو

E element

ئەم ڕێگەیە بەکاردێت بۆ گێڕانەوەی سەر Head ی شارای نۆبە بەست Queue کە بە دیکیوو Deque،پیشان درابێت و درووستکرابێت بەلام کرداری لابردن Remove ئەنجام نادات، سەرە Head یەکەم دانەی پێکهێنەریی ناو دیکیووDeque. ئەگەر دیکیووەکە Deque بەتاڵبێت Empty ئەوا حالەتی نائاساییExceptionی NoSuchElementException ڕوو دەدات .

```
// it will retrive element at first(head) position
  int retval = deque.element();
  System.out.println("Retrieved Element is = " + retval);
```

ئەنجام//

Deque =[]

[20, 30, 20, 30, 15, 25, 11]

Retrieved Element is = 20

گەڕاندنەوەی یەکەم دانەی پێکهێنەر

E get(int index)

ئەم ڕێگەیە بەکاردێت بۆ گەراندنەوەی یەکەم دانەی دیکیوو، ئەگەر دیکیووەکە Deque بەتاڵبێت Empty ئەوا حاڵەتی نائاسایی Exceptionی NoSuchElementException ڕوو دەدات .

```
// print element at index 3
  System.out.println("Element at index 3 :" + linkedlist.get(3));
```

ئەنجام//

Linked List Content: [Hemn, Mala, Karim, Barznji, Software Development Instructor]

Element at index 3 :Barznji

ڕێگەی گەراندنەوەی یەکەم نرخی لیستەی پێکەوە بەستراوو

E getFirst();

ئەم ڕێگەیە یەکەم نرخی دیکیوو Deque دەگێڕێتەوە، بەڵام لای نابات Remove، هیچ داغلکەرێکی Parameter نییە، بەڵام ئەگەر لیستەکە بەتال بوو، ئەوا NoSuchElementException ڕوودەدات.

```
// getFirst() will retrieve element at first(head) position
  int retval = deque.getFirst();
  System.out.println("Retrieved Element is = " + retval);
```

ئەنجام//

```
Deque =[]
[20, 30, 20, 30, 15, 25, 11]
Retrieved Element is = 20
```

رێگەی گەراندنەوەی کۆتا دانه

E getLast()

ئەم رێگەیە کۆتا نرخی دیکیوو Deque دەگێڕێتەوە، بێ ئەوەی لای ببات Remove، هیچ داغڵکەرێکی Parameter نییە، بەڵام ئەگەر لیستەکە بەتال بوو، ئەوا NoSuchElementException ڕوودەدات.

```
 // getLast() will retrieve element at last position
  int retval = deque.getLast();
  System.out.println("Retrieved Element is = " + retval);
   System.out.println("Deque After Retrive Last Elemt"+deque);
```

ئەنجام//

```
Deque =[]
Deque[20, 30, 20, 30, 15, 25, 11]
Retrieved Element is = 11
Deque After Retrive Last Elemt[20, 30, 20, 30, 15, 25, 11]
```

ڕێگەی تاقیکردنەوەی بەتاڵێتی

isEmpty()

ئەم ڕێگەیە Method نرخی ڕاست True دەگێڕێتەوە ئەگەر دیکیوو هیچ دانەیەکی تێدا نەبێت و، نرخی هەڵە False دەگێریتەوە ئەگەر دانەی تیابێت. هیچ داغڵکەرێکی Parameter نییە. حاڵەتی نائاسایش Exception ڕوونادات.

```
boolean retval = deque.isEmpty();
System.out.println("Deque Empty?"+retval);
```

هەروەها دەتوانین لەگەڵ ئەگەر دا بەکاری بهێنین بۆ درووست کردنی گریمانە و چاپ کردن بە گوێرەی گریمانەکە:

```
boolean retval = deque.isEmpty();
  if (retval == true) {
  System.out.println("deque is empty");
  }
  else {
  System.out.println("deque is not empty");
  }
```

بڕووانە هەموو کۆدەکە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
```

```
 */

package deque_implementation_ds_book;

import  java.util.*;
import  java.util.Deque;
/** @author Hemn Barznji   */
public class Deque_Implementation_DS_Book {

    public static void main(String[] args) {
      // create an empty array deque with an initial capacity
   Deque<Integer> deque = new ArrayDeque<Integer>();

          //Print Empty Queue
        System.out.println("Deque ="+ deque);
 boolean retval = deque.isEmpty();
    System.out.println("Deque Empty?"+retval);

     // use add() method to add elements in the deque
   deque.add(20);
   deque.add(30);
```

```
  deque.add(20);

  deque.add(30);

  deque.add(15);

  deque.add(25);

  deque.add(11);


   System.output.println("Deque"+deque);


 retval = deque.isEmpty();

  if (retval == true) {

  System.out.println("deque is empty");

  }

  else {

  System.out.println("deque is not empty");

  }

   }

}
```

ئەنجام::

```
Deque =[]

Deque Empty?true

Deque[20, 30, 20, 30, 15, 25, 11]
```

deque is not empty

ڕێگەی دووبارە کردنەوەی دانەکان لە شوێنی دیاری کراوەوە

ListIterator<E> listIterator()

ئەم ڕێگەیە بەکاردەهێنرێت بۆ گەراندنەوەی لیستێکی دووبارە کراوە List Iterator .

```
// set Iterator at specified index
  Iterator x = deque.iterator();
  // print list with the iterator
  while (x.hasNext()) {
  System.out.println(x.next());
  }
```

یاخود دەتوانین بەم جۆرە بینووسین:

```
// iterator() is used to print all the elements
  // next() returns the next element on each iteration
  System.out.println("printing elements using iterator:");
  for(Iterator itr = deque.iterator();itr.hasNext();)  {
  System.out.println(itr.next());
  }
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
```

Using Iterator

20

30

20

30

15

25

11

زیاد کردنی دانەیەك لە کۆتاییەوە

Boolean offer (E e)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ زیاد کردنی دانەیەکی دیاری کراوو، وەکو دانەی کۆتایی Last Element دیکیوو. ئەگەر دانەیەکی دیاری کراوو بەتاڵ Null بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException پێویستە.

```
// this will insert 50 at the end
  deque.offer(50);
  System.out.println("Deque"+deque);
```

ئەنجام//

[]

Deque[20, 30, 20, 30, 15, 25, 11]

Deque[20, 30, 20, 30, 15, 25, 11, 50]

ڕێگەی زیاد کردنی دانەیەك لەبەردەمەوە

Boolean offerFirst(E e)

ئەم ڕێگەیە بەکاردێت بۆ زیادکردنی دانەیەکی پێکهێنەر Element لەبەردەمەوە ((پێشەوە)) Front، . ئەگەر دانە دیاری کراووەکە بەتاڵ Null بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException پێویستە.

```
// this will insert 10 at the front
   deque.offerFirst(10);
   System.out.println("Deque"+deque);
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
Deque[10, 20, 30, 20, 30, 15, 25, 11]
```

ڕێگەی زیاد کردنی دانەیەك لە کۆتاییەوە

Boolean offerLast(E e)

ئەم ڕێگەیە بەکاردێت بۆ زیادکردنی دانەیەکی پێکهێنەر Element لە کۆتاییەوە ، ئەگەر دانە دیاری کراووەکە بەتاڵ Null بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException پێویستە.

```
// this will insert 40 at the end
```

```
  deque.offerLast(40);

  System.out.println("Deque"+deque);
```

ئەنجام//

```
[]

Deque[20, 30, 20, 30, 15, 25, 11]

Deque[20, 30, 20, 30, 15, 25, 11, 40]
```

ڕێگەیی گەڕاندنەوەی نرخی سەرە

E Peek ()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی نرخی دانەی سەرە Head ، ی کیوو Queue کە بە دیکیوو Deque دروستکرابێت. یان نرخی بەتاڵ Null دەگێڕێتەوە ئەگەر دیکیووەکە بەتاڵ بێت.

```
// this will retrieve head of the queue

  int retval = deque.peek();

  System.out.println("Retrieved Element is " + retval);
```

ئەنجام//

```
[]

Deque[20, 30, 20, 30, 15, 25, 11]

Retrieved Element is 20
```

ڕێگەیی گەڕاندنەوەی نرخی یەکەم دانە

E PeekFirst ()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی نرخی یەکەم دانەی دیکیوو Deque، یان گەڕاندنەوەی نرخی بەتال Null ئەگەر لیستەکە بەتاڵ Empty بوو. بەڵام دانەکە لانابات Remove.

```
  // this will retrieve head of the queue
 int retval = deque.peekFirst();
 System.out.println("Retrieved Element is " + retval);
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
Retrieved Element is 20
```

ڕێگەیی گەڕاندنەوەی نرخی کۆتا دانە

E PeekLast ()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی نرخی کۆتا دانەی دیکیوو Deque ((واتە کلك Tail))، یان گەڕاندنەوەی نرخی بەتال Null ئەگەر لیستەکە بەتاڵ Empty بوو. بەڵام دانەکە لانابات Remove.

```
// this will retrieve end value
 int retval = deque.peekLast();
 System.out.println("Retrieved Element is " + retval);
```

ئەنجلم//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
```

Retrieved Element is 11

ڕێگەی گەڕاندنەوە و لابردنی سەرە

E Poll()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوە و لابردنی سەرە Head ((یەکەم دانەی پێکهێنەریی)) کیوو Queue کە بە دیکیوو Deque دروستکرابێت، هەروەها Null دەگێڕێتەوە ئەگەر بەتاڵ Empty بێت.

```
int retval = deque.poll();
  System.out.println("Element removed is " + retval);
```

ئەنجام//

[]

Deque[20, 30, 20, 30, 15, 25, 11]

Element removed is 20

ڕێگەی گەڕاندنەوە و لابردنی یەکەم دانە

E PollFirst()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوە و لابردنی یەکەم دانەی پێکهێنەرییHead دیکیوو Deque. یان گەڕاندنەوەی نرخی هیچ شتێك Null ئەگەر لیستەکە بەتاڵ Empty بوو.

```
int retval = deque.pollFirst();
```

```
System.out.println("Element removed is " + retval);
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
Element removed is 20
```

ڕێگەی گەڕاندنەوە و لابردنی کۆتا دانە

E pollLast()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوە و لابردنی کۆتا دانەی پێکهێنەریی دیکیوو Deque. یان گەڕاندنەوەی نرخی هیچ شتێك Null ئەگەر لیستەکە بەتاڵ Empty بوو.

```
int retval = deque.pollLast();
System.out.println("Element removed is " + retval);
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
Element removed is 11
```

ڕێگەی دەرهێنانەوەی دانەی پێکهێنەر

E pop()

ئەم ڕێگەیە بەکاردەهێنرێت بۆ دەرهێنانەوەی Pop دانەی پێکهێنەر Element لە ریزەخانەی ستوونی ((ریزە ڕەفە)) Stack دا، کە بە دیکیوو Deque درووستکرابێت.

```
int retval = deque.pop();
```

```
System.out.println("Element removed is " + retval);
```

ئەنجام//

```
Deque[20, 30, 20, 30, 15, 25, 11]
Element removed is 20
```

ڕێگەی تیادانی دانەی پێکهێنەر

Void push(E e)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ کرداریی دانان و خستنە ناوی push دانەی پێکهێنەر بۆ ڕیزەخانە ((شارای ڕەفەیی)) ی Stack کە بە دیکیوو Deque درووستکرابێت، هەروەها هیچ جۆرە نرخێك ناگێرێتەوە. . ئەگەر دانە دیاری کراووەکە بەتاڵ Null بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException پێویستە.

```
// adding elements using push() method
  deque.push(10);
  deque.push(15);
System.out.println("Deque"+deque);
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
Deque[15, 10, 20, 30, 20, 30, 15, 25, 11]
```

ڕێگەی گەڕاندنەوە و لابردنی سەرە

E Remove()

ئەم ڕێگەیە بەکاردەهێنرێت بۆ گەڕاندنەوە Retrieve و لابردنی Remove سەرە Head ((یەکەم دانە ی پێکهێنەر First Element)) ی کیوو Queue کە بە دیکیوو Deque درووستکرابێت. هیچ داغڵکەرێك Parameter ی نییە. ئەگەر دیکیووەکە Dequeبەتاڵ بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException ڕوودەدات.

```
// this will remove element at the first(head) postion
   int retval = deque.remove();
   System.out.println("Element removed is: " + retval);
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
Element removed is: 20
```

ڕێگەی لابردنی دانەیەکی پێکهێنەر لە یەکەم دەرکەوتن دا

Boolean remove(Object o)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لابرنی دانەیەکی پێکهێنەر ئەگەر دانەکە هەبوو لای دەبات و نرخی ڕاست True دەگێڕێتەوە.

```
// deque contains element 30, returns true
   boolean retval = deque.remove(30);
```

بڕووانە هەموو کۆدەکە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
```

```
 * Software Development Instructor
 */

package deque_implementation_ds_book;

import  java.util.*;
import  java.util.Deque;
/** @author Hemn Barznji   */
public class Deque_Implementation_DS_Book {

    public static void main(String[] args) {
       // create an empty array deque with an initial capacity
   Deque<Integer> deque = new ArrayDeque<Integer>();



    System.out.println(deque);

// use add() method to add elements in the deque
   deque.add(25);
   deque.add(30);
   deque.add(20);
```

```
deque.add(40);

// printing all the elements available in deque
for (Integer number : deque) {
System.out.println("Number = " + number);
}

// deque contains element 30, returns true
boolean retval = deque.remove(30);
if (retval == true) {
System.out.println("element 30 is removed from the deque");
}
else {
System.out.println("element 30 is not contained in the deque");
}

// deque does not contain element 15, returns false
retval = deque.remove(15);
if (retval == true) {
System.out.println("element 15 is removed from the deque");
}
```

```
  else {
  System.out.println("element 15 is not contained in the deque");
  }

  // printing remaining elements
  for (Integer number : deque) {
  System.out.println("Number = " + number);
  }
   }
}
```

ئەنجام::

```
[]
Number = 25
Number = 30
Number = 20
Number = 40
element 30 is removed from the deque
element 15 is not contained in the deque
Number = 25
Number = 20
Number = 40
```

رێگەیی لابردنی یەکەم دانە

E removeFirst()

ئەم رێگەیە بەکاردەهێنرێت بۆ لابردن و گەراندنەوەی یەکەم دانەی پێکهێنەر Head لە دیکیوو Dequeدا. ئەگەر دیکیووەکە Dequeبەتاڵ بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException روودەدات.. هیچ داغڵکەرێکی نییە No Parameter.

```
// this will remove element at the first(head) postion
  int retval = deque.removeFirst();
  System.out.println("Element removed is: " + retval);
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
Element removed is: 20
```

رێگەی لابردنی دانەیەکی پێکهێنەر لە یەکەم دەرکەوتن دا

Boolean removeFirstOccurrence(Object o)

ئەم رێگەیە بەکاردەهێنرێت بۆ لابرنی دانەیەکی پێکهێنەر لەیەکەم دەرکەوتنی دا، چونکە لە وانەیە هەمان دانە چەند جارێکی تر لە لیستەکەدا هاتبێت و، لە شوێنی جیاوازدا، دەرکەوتبێت، ئەگەر دانەکە هەبوو لای دەبات و نرخی ڕاست True دەگێرِێتەوە.

```
// this will remove first occurrence of element 20
```

```
  deque.removeFirstOccurrence(20);
    System.out.println("Deque"+deque);
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
Deque[30, 20, 30, 15, 25, 11]
```

ڕێگەیی لابردنی کۆتا دانە

E removeLast()

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لابردن و گەراندنەوەی کۆتا دانەی پێکهێنەر لە دیکیوو Dequeدا. ئەگەر دیکیووەکە Dequeبەتاڵ بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException ڕوودەدات.. هیچ داغڵکەرێکی نییە No Parameter.

```
// this will remove element at last position
  int retval  = deque.removeLast();
  System.out.println("Element removed is: " + retval);
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
Element removed is: 11
```

ڕێگەی لابردنی دانەیەکی پێکهێنەر لە دواین دەرکەوتن دا

Boolean removeLastOccurrence(Object o)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لابرنی دانەیەکی پێکهێنەر لە دووایین دەرکەوتنی دا، چونکە لە وانەیە هەمان دانە چەند جارێکی تر لە لیستەکەدا هاتبێت و، لە شوێنی جیاوازدا، دەرکەوتبێت، ئەگەر دانەکە هەبوو لای دەبات و نرخی ڕاست True دەگێڕێتەوە.

```
// this will remove element at last Occurence
boolean retval  = deque.removeLastOccurrence(20);
System.out.println("Element removed is: " + retval);
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
Element removed is: true
```

ڕێگەیی گەڕاندنەوەی نرخی ژمارەیی دانەکانی دیکیوو

Int size()

ئەم ڕێگەیە بەکاردێت بۆ زانینی حەجمی دیکیوو Deque Size کە دەکاتە ژمارەی دانە پێکهێنەرەکانی ناو دیکیوو. هیچ داغڵکەریکی Parameter نییە.

```
// print the size of the Deque
  System.out.println("List size:" + deque.size());
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]
List size:7
```

لە بەرگرتنەوەی دیکیوو بۆ ناو Array

Object[] toArray()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی ئەرەیی Array کە هەموو دانەکانی دیکیووی تیابێت بە ڕێکخستنی لە یەکەمەوە First بۆ کۆتایی Last،

```
 // printing the array length of elements added above
 Object[] ob = deque.toArray();
 System.out.println("\n Array Size : " + ob.length);
 // print the array
 for (int i = 0; i < deque.size(); i++)
{
 System.out.println("Array:" + ob[i]);
}
```

ئەنجام//

```
[]
Deque[20, 30, 20, 30, 15, 25, 11]

 Array Size : 7
Array:20
Array:30
Array:20
Array:30
```

Array:15

Array:25

Array:11

پرسیار

پرسیار: شارای ڕەفەیی Stack درووستبکە بە بەکارهێنانی دیکیوو Deque.

پرسیار: شارای نۆبە بەستن Queue درووست بکە بە بەکارهێنانی دیکیوو Deque.

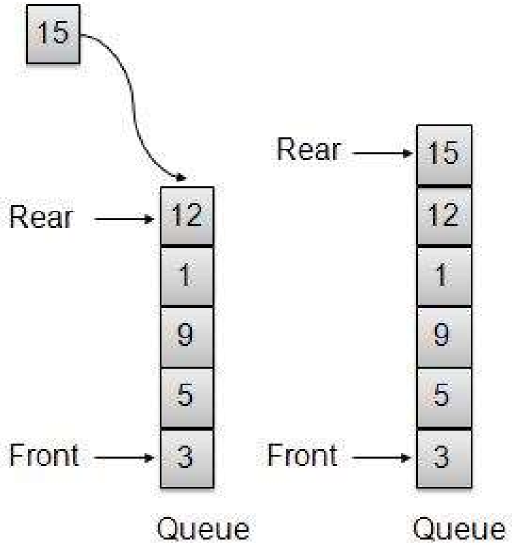
پرسیار : دکیوویەك درووست بکە کە لە دوواوە Rear((کلك Tail)) تەنها بتوانی دانە داغلبکەی، بەلاّم لە پێشەوە Front ((سەر Head)) بتوانی دانە بسریتەوە و دانەش داغلبکەی.

پرسیار: دیکیوویەك درووست بکە، بە جۆرێك لە دوواوە Rear((کلك Tail)) بتوانی دانە داغلبکەی یان بیسریتەوە، بەلاّم لە پێشەوە Front ((سەر Head)) تەنها بتوانی دانە بسریتەوە.

شارای نۆبەی

پێشخراوو

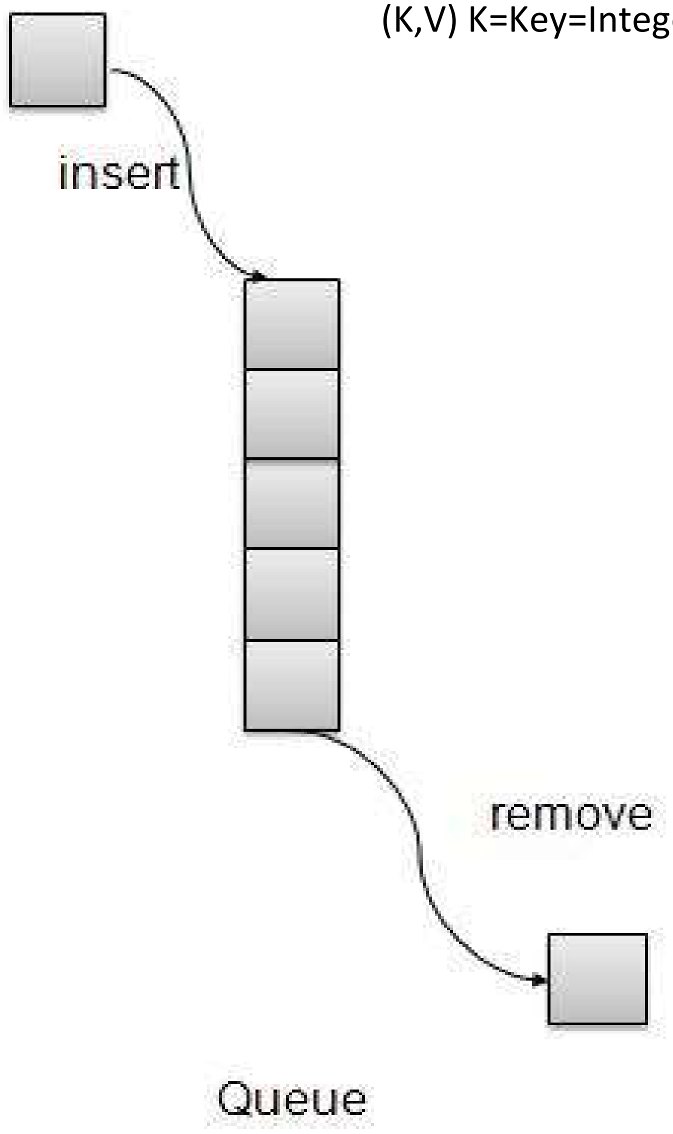
Priority Queue



One item inserted at rear end

شارای نۆبەیی پێشخراوو Priority Queue شێوازێکی پێکهاتنی زانیاری تایبەتە بە بەراورد بە شارای نۆبە بەستن Queue. شارای نۆبەیی پێشخراو وەکو کیوویی ئاسایی وایە لە ڕووی بوونی ڕێگەکانەوە Same Method، بەڵام بە جیاوازی زۆرەوە، لە شارای نۆبەیی پێشخراوو Priority Queueدا، دانەکان ڕێکدەخرێن بە هۆی نرخی کلیلێکەوە Key Value، بە جۆرێك کە نزمترین نرخی ئەو کلیلە لە پێشەوە Front دەبێت و، بەرزترین نرخی کلیلەکە Highest Value of Key لە دوواوە Rear دەبێت. یان بە پێچەوانەشەوە راستە. کەواتە پێشخراوویی دانەکان لەسەر بنچینەی نرخی کلیلەکەیە Key Value، بە جۆرێك کە نرخەکەی نزمترە Lower the Value، پێشخراوییەکەی بەرزترە Higher the Priority.

(K,V) K=Key=Integer …. V=Value



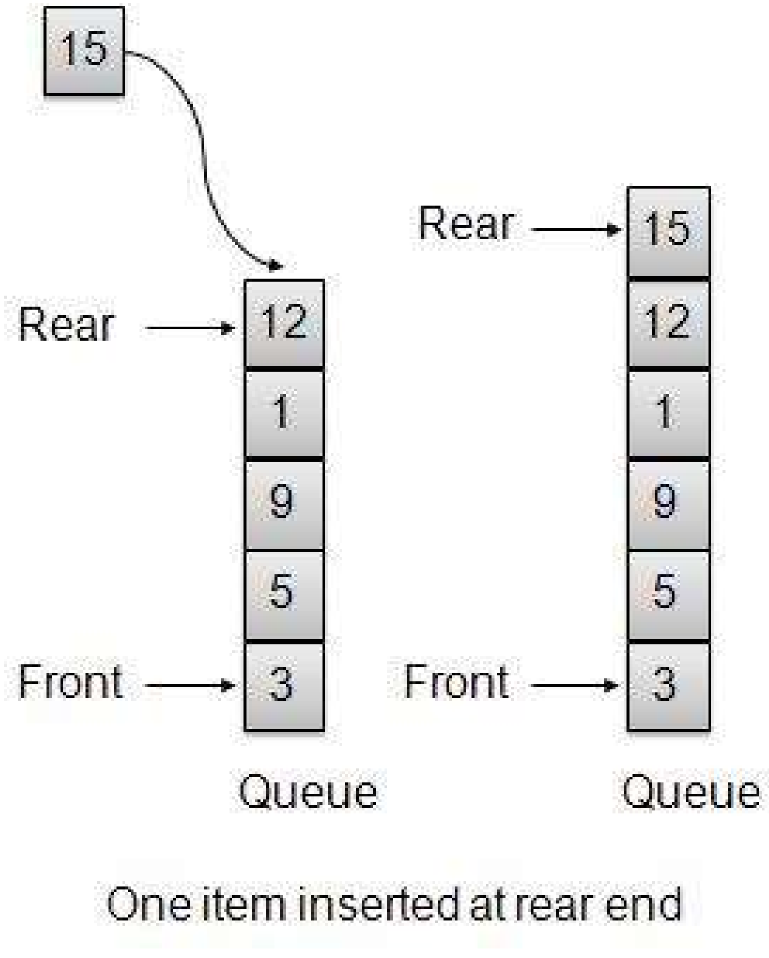
کرداره بنچینەییەکان

Basic Operations

بە شێوەیەکی گشتی 5 کرداری گرنگ و سەرەکیمان هەیە، کە لە شارای نۆبەیی پێشخراوودا، جێبەجێدەبێت و، بەکورتی باسیان دەکەین بە وێنەوە.

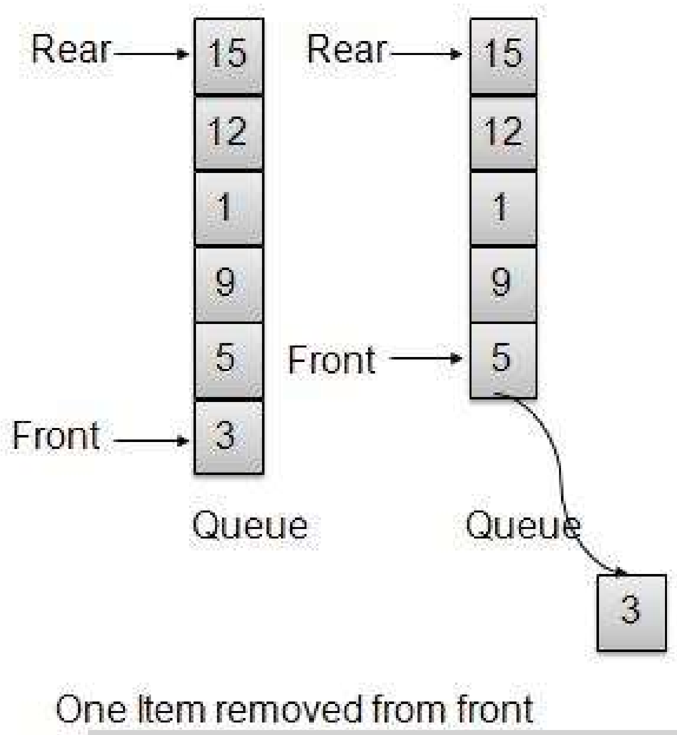
خستنە نۆبەوە ((خستنە سەرەوە)) Enqueue یان داغڵکردن Insert

زیاد کردنی دانەیە بۆ بەشی دوواوەی Rear کیووەکە. شارای نۆبەیی پێشخراوو Priority Queue دانەکان داغڵدەکات بە گوێرەی ڕێکخستن و سیستەم، بە جۆرێك کە زانیاری Data لەگەڵ نرخێکی بەرز High Value پێشخراووییەکی نزمی Low Priority هەیە.



دەرکردن لە نۆبەوە ((لەسەرە هێنانە دەرەوە)) Dequeue یان لابردن Remove

لابردن و سرینەوەی Remove دانەیە لە بەشی پێشەوەی Front شارای نۆبەیی پێشخراوەوە Priority Queue،



کردارەکانی تر Other Operations

- گەڕاندنەوەی نرخی سەرە Peak () واتە نرخی پێشەوە Front ی شارای نۆبەی پێشخراوو.
- پڕە isFull ()، پشکنینی شارای نۆبەیی پێشخراوو ئەگەر پربێت، یان نا.
- بەتاڵە isEmpty ()، پشکنینی شارای نۆبەی پێشخراوو ئەگەر بەتاڵ بێت.

Example نموونە

(K, V) K=Key=Integer ….. V=Value=Letter

- شارای نۆبەی پێشخراووی بەتاڵ Empty Priority Queue درووست دەکەین.

PQ={ };

- نرخی 5، ئەی داغڵ دەکەین.

Insert (5,A) …..> PQ={ (5,A) };

- نرخی نۆ، سی داغڵدەکەین.

Insert (9,C) ……> PQ={ (5,A), (9,C), };

- نرخی سێ، بی داغڵدەکەین.

Insert (3,B)…..> PQ={ (5,A), (9,C), (3,B) };

- نرخی حەوت، دی داغڵدەکەین.

Insert (7,D)….> PQ={ (5,A), (7,D), (9,C), (3,B)};

- نرمترین و بچوکترین نرخی کلیل دەکات سێ، بۆیە سێ بی لە پێشەوە Frontدایە:

Min() …..> Return (3,B);

- لابردنی نرخی بچووك ئەنجام دەدەین:

RemoveMin() …> PQ={ (5,A), (7,D), (9,C) };

- دەمانەوێ حەجمی شارای نۆبەیی پێشخراوو بزانین:

Size() ….> Return 3

- لابردنی بچوکترین:

RemoveMin()….> Return (5.A) PQ={ (7,D), (9,C)};

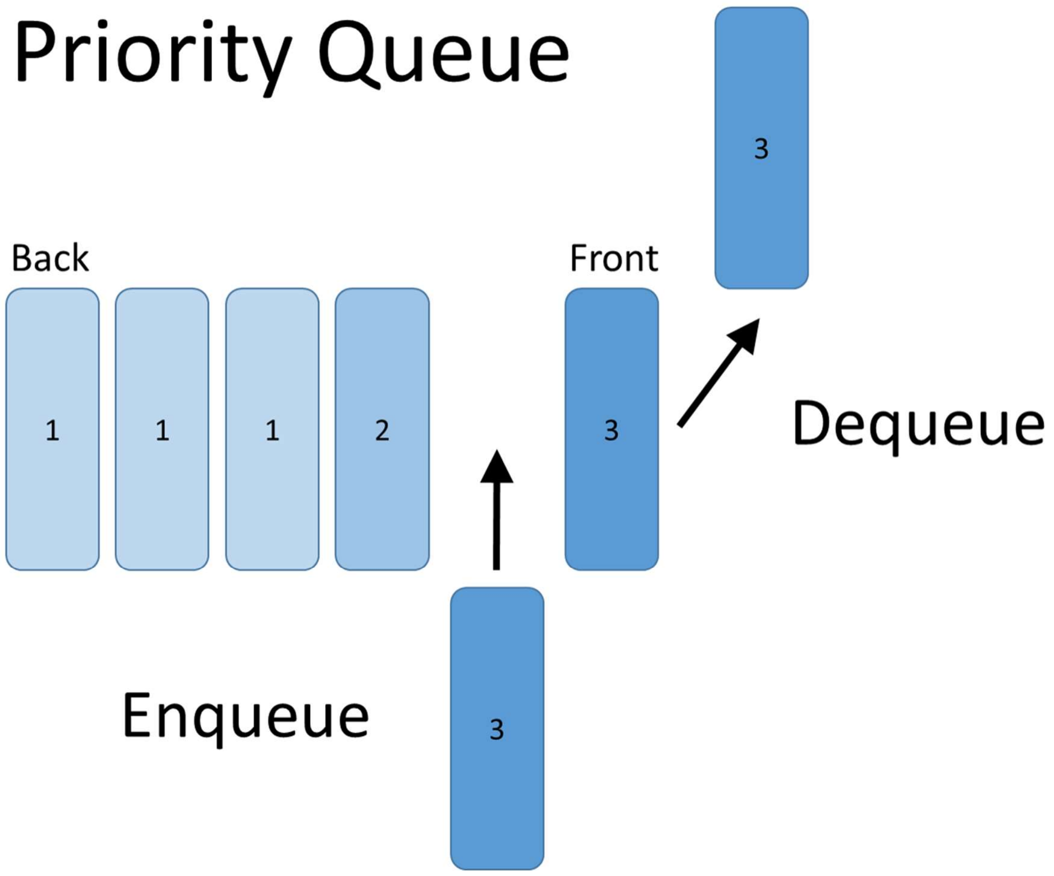
- لابردنی نرخی بچوکترین:

RemoveMin()…> Return (7,D) PQ={ (9,C) };

- لابردنی نرخی بچووك:

RemoveMin ….> Return (9,C)   PQ={ };

نموونەیەکی وێنەیی



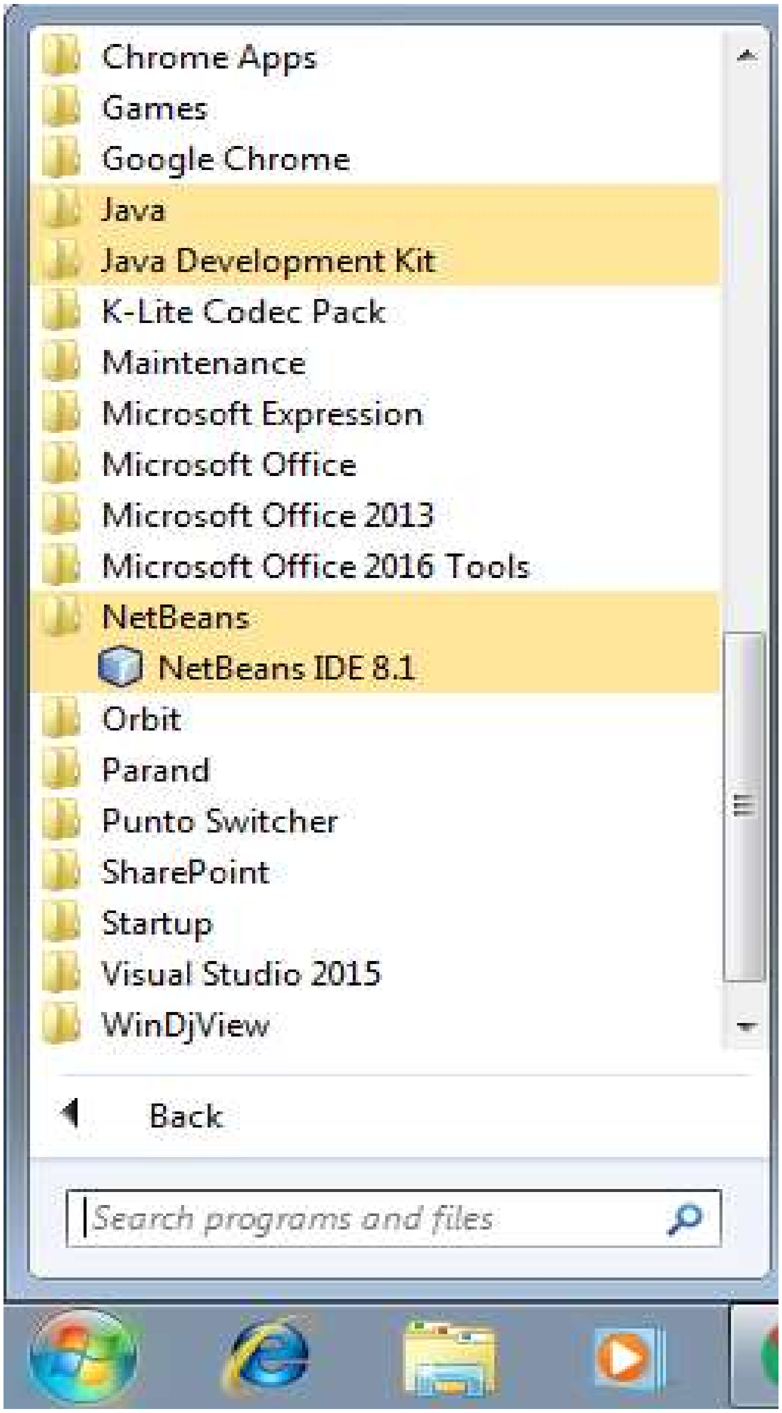
جێبەجێکردنی شارای نۆبەی پێشخراوو

Priority Queue Immplementation

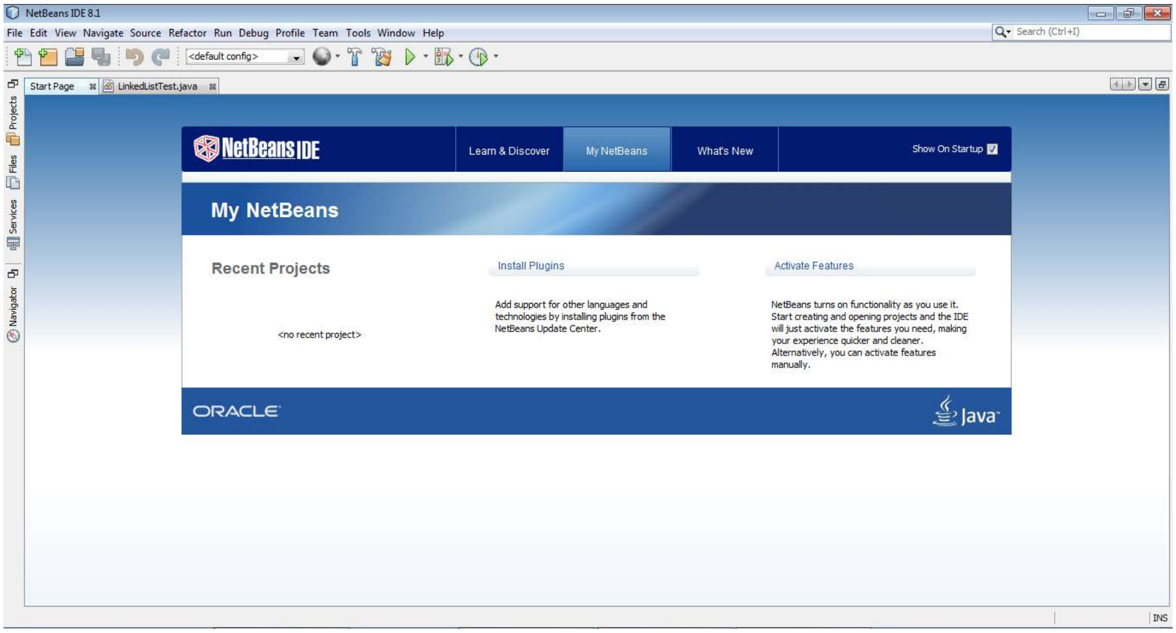
جێبەجێکردنی شارای نۆبەیی پێشخراوو

Priority Queue Implementation

1-بەرنامە نێتبین NetBeans دەکەینەوە، بە کلیك کردن لەسەر دووگمەی دەستپێکردن Start Button، پاشان کلیك لەسەر NetBeans و دوواتر کلیك لەسەر NetBeans IDE 8.1:

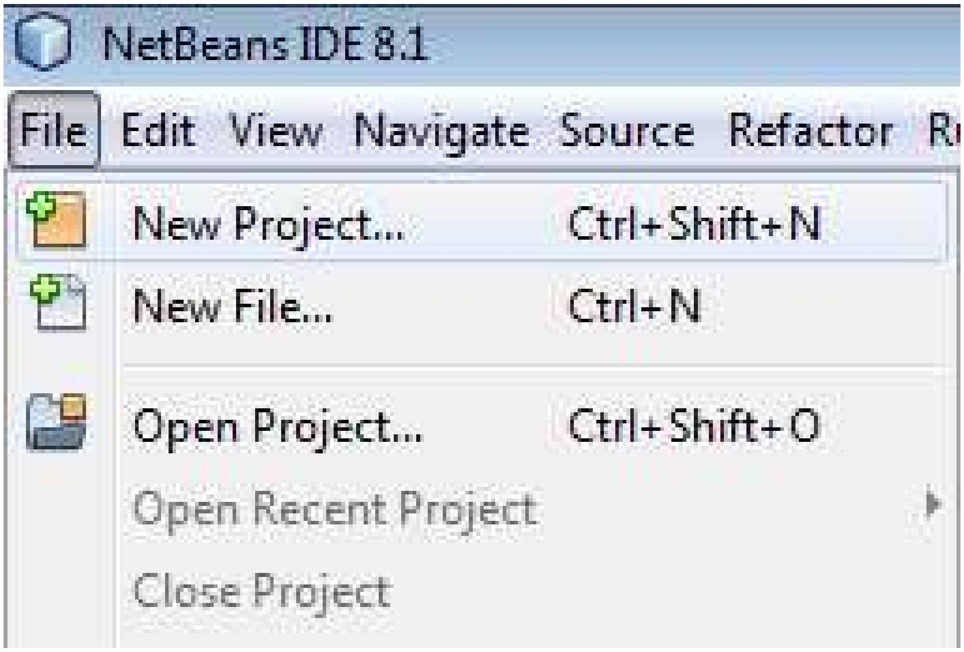


2-ڕووكاری بەرنامەی نێتبینز NetBeansدەكرێتەوە.

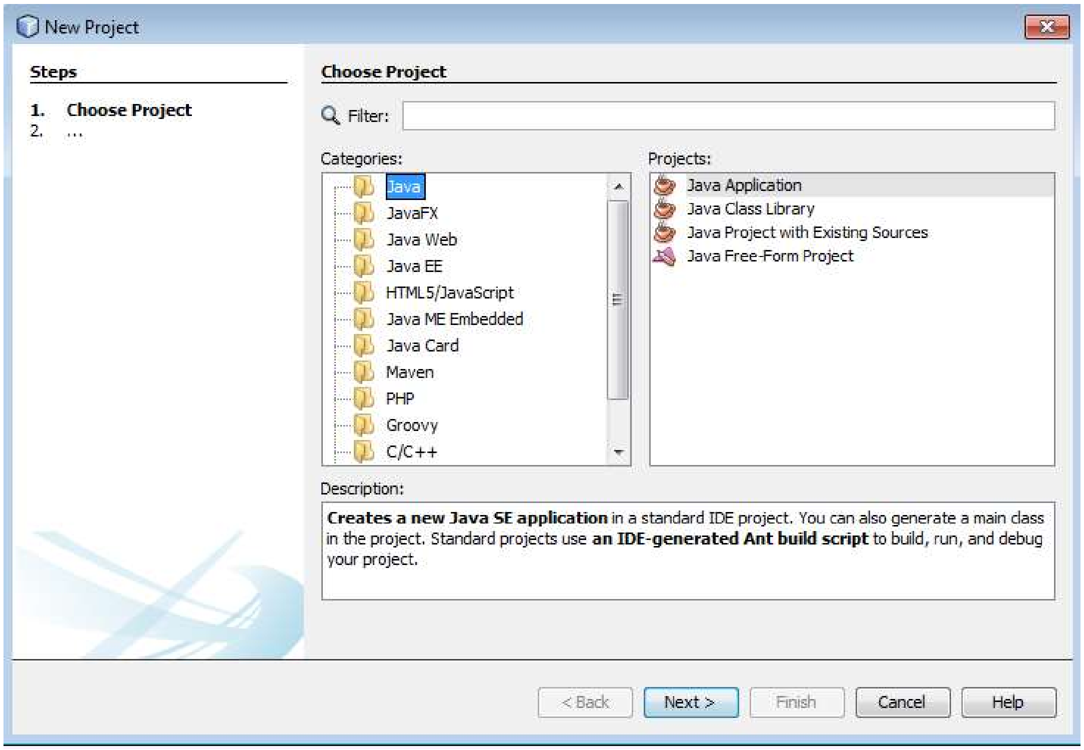


3-كلیك لەسەر دووگمەی فایل Fileدەكەین، بۆ ئەوەی پرۆژەیەكی نوێ درووست بكەین.

4-كلیك لەسەر پرۆژەی نوێ New Project دەكەین:



5-ڕووکاری تایبەت بە پرۆژەی نوێ New Projectدەکرێتەوە و، کلیك لەسەر دووگمەی دوواتر Next دەکەین.



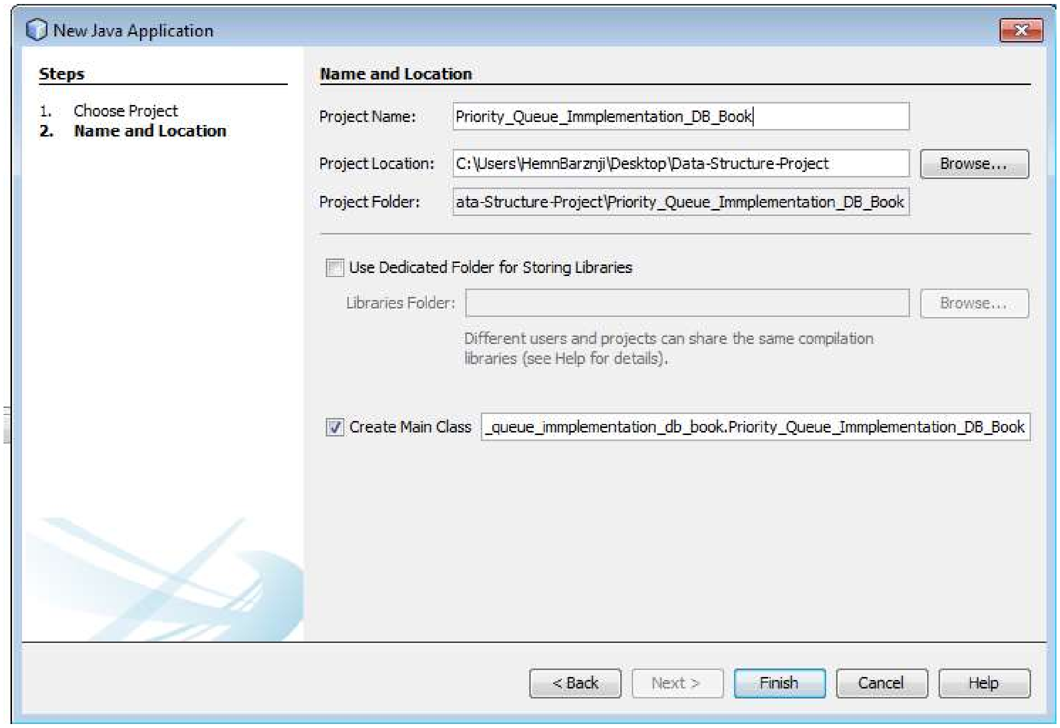
6-ڕووکارێکی نوێ دەکرێتەوە:

-لەبەشی ناوی پرۆژە Project Nameدا، ناوێك بۆ پرۆژەکە بنووسە.

-لەبەشی شوێنی پرۆژە Project Locationدا، شوێنی پرۆژەکە دیاری بکە.

-کلیك لەناو سنووقی پشکنینی Check Box درووستکردنی پۆلی سەرەکی Create Main Class بکە، ئەگەر دەتەوێت پۆلێکت بۆ درووست بکات بە ناوێکەوە کە بەرنامەکە دیاری دەکات.

-لە دووای ئەو هەنگاوانە کلیك لەسەر دووگەی کۆتای هاتن Finishبکە.



7-ئیستا پرۆژەکە درووست بوو:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
package priority_queue_immplementation_db_book;

/** @author HemnBarznji    */
public class Priority_Queue_Immplementation_DB_Book {

    public static void main(String[] args) {
        // TODO code application logic here
    }

}
```

8-بەمەش پۆلێك Class درووست بوو، کە ڕێگەی سەرەکیشی Main Class تیایە.

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
package priority_queue_immplementation_db_book;


/** @author HemnBarznji   */
public class Priority_Queue_Immplementation_DB_Book {


    public static void main(String[] args) {
        // TODO code application logic here
    }


}
```

کۆد نووسی بۆ شارای نۆبەیی پێشخراوو

Coding for Deque

لە زۆربەی زمانەکانی بەرنامە سازیی دا، پۆلی تایبەت و رێگەی دیاری کراوو ئامادەکراوە بۆ جێبەجێکردنی شارای نۆبەیی پێشخراوو Priority Queue. زمانی بەرنامە سازیی جاڤاش کە لەم کتیبەدا بەکارمان هێناوە، پۆلی شارای نۆبەیی پێشخراووی دابین کردووە و بەکاری دەهێنین.

1-سەرەتا پاکەتی جاڤا یوتیلْ Java Util دەهێنینە ناوەوە:

```
import  java.util.*;
```

2-ڕاگەیاندن و بڵاوکردنەوەی شارای نۆبەیی پێشخراوو ئەنجام دەدەین، لەناو رێگەیی سەرەکی Main Method دا، بۆ مەبەستی درووستکردنی شارای نۆبەی پێشخراوی بەتاڵ، بڕوانە کۆدەکەی خوارەوە:

```
// create priority queue
PriorityQueue < Integer >  prq = new PriorityQueue < Integer > () ;
```

شارای نۆبە گرتن لە هەردوو لاوە Deque چاپ Print دەکەین کە بەتاڵە و هیچی تیا نییە. چونکە هیچ دانەیەکمان داغلْ نەکردووە بۆی.

```
System.out.println(prq);
```

3-پاشان کرداری داغڵکردنی زانیاری ئەنجام دەدەین، بە هۆیی رێگەیی زیاد کردنەوە:

```
prq.add(20);
```

4-بەهەمان شێوە هەموو دانەکانی تر زیاد دەکەین کە دەمانەوێت:

```
prq.add(15);
prq.add(25);
prq.add(10);
```

5-ڕێگەی پیشاندان Display Method دەنووسین، بۆ چاپکردن:

```
//Print After Adding
 System.out.println(prq);
```

6-ڕێگەی حەجم size() بەکاردەهێنین بۆ زانینی حەجم:

```
System.out.println("Size: " + prq.size());
```

7-دەتوانین ڕێگەی لابردن remove() بە کاربهێنین بۆ لابردنەوەی دانەکان لە ناو شارای نۆبەی پێشخراوو Priority Queue:

```
  // Remove Item
  prq.remove();
   //Print After Remove
    System.out.println(prq);
```

ئەمەش هەموو کۆدەکەیە پێکەوە:

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
package priority_queue_immplementation_db_book;
import java.util.*;
/** @author HemnBarznji   */
public class Priority_Queue_Immplementation_DB_Book {
```

```
 public static void main(String[] args) {
    // create priority queue
  PriorityQueue < Integer >  prq = new PriorityQueue < Integer > () ;
  System.out.println("Priority Queue = "+prq);

  prq.add(20);
  prq.add(15);
  prq.add(25);
  prq.add(10);

  //Print After Adding
    System.out.println(prq);
System.out.println("Size: " + prq.size());
// Remove Item
  prq.remove();

  //Print After Remove
    System.out.println(prq);
 // Remove Item
  prq.remove();
```

```
  //Print After Remove
    System.out.println(prq);

    // Remove Item
  prq.remove();

  //Print After Remove
    System.out.println(prq);
  }
}
```

8-کلیك لەسەر دووگمەی جێبەجێکردنی پرۆژە Run Projectدەکەین:

9-بەرهەم و ئەنجام::

```
Priority Queue = []
[10, 15, 25, 20]
Size: 4
[15, 20, 25]
[20, 25]
[25]
```

دروستکەرەکانی پۆل

Class Constructors

پۆلی شارای نۆبەیی پێشخراوو Priority Queue کە ناوی PriorityQueue ، چەند دروستکەرێکی Constructor هەیە، کە ئەمانەی لای خوارەوەن:

- دروستکەری() PriorityQueue

ئەم دروستکەرە Constructor بەکاردێت بۆ دروستکردنی شارای نۆبەیی پێشخراووی بەتاڵ Empty Priority Queue، بە توانایەکی سەرەتایی تەواو بۆ 16 دانەی پێکهێنەر.

- دروستکەری PriorityQueue(Collection<? extends E> c)

ئەم دروستکەرە Constructor بەکاردێت بۆ دروستکردنی شارای نۆبەیی پێشخراوو Priority Queue، کە دانە پێکهێنەرەکانی کۆکراوەیەکی دیاری کراوی Specified Collection تیابێت.

- دروستکەری PriorityQueue(int initialCapacity)

ئەم دروستکەرە Constructor بەکاردێت بۆ دروستکردنی شارای نۆبەیی پێشخراوو Priority Queue، لەگەڵ دیاری کردنی توانای سەرەتایی بۆ هەڵگرتنی دانە پێکهێنەرەکان.

- دروستکەری PriorityQueue(int initialCapacity, Comparator<? super E> comparator)

ئەم دروستکەرە Constructor بەکاردێت بۆ دروستکردنی شارای نۆبەیی پێشخراوو Priority Queue، لەگەڵ دیاری کردنی توانای سەرەتایی بۆ هەڵگرتنی دانە پێکهێنەرەکان کە دانەکانی ڕێکدەخرێت بە گوێرەی بەراوردکردنێکی دیاری کراوو Specified Comparator.

- دروستکەری PriorityQueue(PriorityQueue<? extends E> c)

ئەم دروستکەرە Constructor بەکاردێت بۆ درووستکردنی شارای نۆبەیی پێشخراوو Priority Queue، کە دانەکانی ناوی لە شارایەکی نۆبەیی پێشخراوی ترەوەیە Priority Queue.

- دروستکەری PriorityQueue(SortedSet<? extends E> c)

ئەم دروستکەرە Constructor بەکاردێت بۆ درووستکردنی شارای نۆبەیی پێشخراوو Priority Queue، کە دانەکانی ناوی لە کۆمەڵەیەکی ڕێکخراوەوەیە Sorted Set.

ڕێگەکان Methods

بەهەمان شێوەی درووستکەرەکان Constructor، ڕێگەکانیش Methods دەبێت بناسێنرێن و باس بکرێن بۆ ئەوەی بە گوێرەی پێویست کەڵکیان لێوەربگیرێت و بەکاربهێنرێن. ژمارەی ڕێگە ئامادەکراوەکان 12 دانەیە، کە بە ئاسانی دەتوانین بەکاریان بهێنین بۆ ئەنجامدانی کارەکانمان.

هەموو ڕێگەکان لەسەر نموونەکەی سەرەوە جیبەجێدەکەین بۆ ئەوەی کەمترین کۆد بنووسینەوە، بەلام لەهەر شوێنێکدا بە پێویستمزانی هەموو کۆدەکە دەنووسمەوە.

ڕێگەی زیادکردنی دانەپێکهێنەر لە جۆریی بولییەن

Boolean add(E e);

ئەم ڕێگەیە Method بەکاردێت بۆ سەربارکردن و زیادکردنی دانەیەکی پێکهێنەر بۆ ناو شارای نۆبەیی پێشخراوو Priority Queue. ئەگەر دانەیەکی دیاری کراوو بەتاڵ Null بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException پێویستە. بەڵام ئەگەر دانەیەکی دیاری کراوو نەتوانرێت بەراورد بکرێت لەگەڵ دانەی ناو شارای نۆبەیی پێشخراوو ئەوا حاڵەی نا ئاسا Exception ی ClassCastException ڕوودەدات.

نموونە:

```
prq.add(20);
```

```
prq.add(15);
prq.add(25);
prq.add(10);
//Print After Adding
  System.out.println("Priority Queue = "+prq);
```

ئەنجام//

```
Priority Queue = []
Priority Queue = [10, 15, 25, 20]
```

سڕینەوە

Void clear();

ئەم ڕێگەیە بەکاردەهێنرێت بۆ لابردنی Remove هەموو دانەکانی ناو شارای نۆبەی پێشخراوو Priority Queue و. ئەم ڕێگەیە هیچ نرخێك ناگێڕێتەوە.

```
 // clear the queue
prq.clear();
System.out.println ( "Priority queue values after clear: " + prq);
```

ئەنجام//

```
Priority Queue = []
Priority Queue = [10, 15, 25, 20]
Priority queue values after clear: []
```

ڕێگەی بەراورد کردن

Comparator<? super E> comparator()

ئەم ڕێگەیە بەراورد کردن دەگێڕێتەوە کە بەکاردێت بۆ ڕێکخستنی دانەکان Element Order لە شارای نۆبەیی دا، یاخود بەتاڵ ((هیچ شتێك)) Null دەگێڕێتەوە ئەگەر شاراکە ڕێکخرابێت بە گوێرەی ڕێکخستنی سرووشتی دانەکان.

```
// create comparator
Comparator comp = prq.comparator();
System.out.println ( "Comparator value is: "+ comp);
```

ئەنجام::

```
Priority Queue = []
Priority Queue = [10, 15, 25, 20]
Comparator value is: null
```

تیابوون

Boolean contain (object O)

ئەم ڕێگەیە نرخی ڕاست True دەگێڕێتەوە، لە حاڵەتی هەبوونی دانەیەکی دیاری کراوویی پێکهێنەر لە شارای نۆبەیی پێشخراوو Priority Queue دا، داغڵکەری ئۆ O ئەو دانە پێکهێنەرەیە کە تاقیدەکرێتەوە و، بۆ ئەوەی بزانین هەیە؟ یان نییە لە لیستەکەدا. ئەگەر هەبێت True دەگێڕێتەوە، ئەگەر نەبێت False دەگێڕێتەوە.

```
// check if queue contains 5
boolean b = prq.contains(5);
```

```
System.out.println ( "Priority queue contains 5: " + b);
```

ئەنجام//

```
Priority Queue = []
Priority Queue = [10, 15, 25, 20]
Priority queue contains 5: false
```

ڕێگەیی دووبارە کردنەوەی پێچەوانەیی

Iterator<E> iterator()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی دانە پێکهێنەرەکانی شارای نۆبەیی پێخراوو Priority Queue.

```
// create iterator from the queue
  Iterator it = prq.iterator();
    System.out.println("Priority Queue = "+prq);
```

دەتوانین بەم جۆرەش بینووسین:

```
// create iterator from the queue
  Iterator it = prq.iterator();
   System.out.println ( "Priority queue values are: ");
  while (it.hasNext()){
  System.out.println ( "Value: "+ it.next()); }
```

ئەنجام//

Priority Queue = []

Priority Queue = [10, 15, 25, 20]

Priority queue values are:

Value: 10

Value: 15

Value: 25

Value: 20

زیاد کردنی دانەیەك لە کۆتاییەوە

Boolean offer (E e)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ زیاد کردنی دانەیەکی دیاری کراوو، بۆ شارای نۆبەیی پێشخراوو Priority Queue . ئەگەر دانەیەکی دیاری کراوو بەتاڵ Null بێت، ئەوا حاڵەتی نائاسا Exception ی NullPointerException پێویستە. . بەڵام ئەگەر دانەیەکی دیاری کراوو نەتوانرێت بەراورد بکرێت لەگەڵ دانەی ناو شارای نۆبەیی پێشخراوو ئەوا حاڵەی نا ئاسا Exception ی ClassCastException ڕوودەدات.

```
// add using offer() function call
  prq.offer(122);
  System.out.println ( "Priority queue values after addition: "+ prq);
```

ئەنجام//

Priority Queue = []

Priority Queue = [10, 15, 25, 20]

Priority queue values after addition: [10, 15, 25, 20, 122]

رێگەیی گەراندنەوەی نرخی سەرە

E Peek ()

ئەم رێگەیە بەکاردێت بۆ گەراندنەوەی نرخی دانەی سەرە Head ، ی کیووی پێشخراوو Priority Queue . یان نرخی بەتاڵ Null دەگێرێتەوە ئەگەر شارای نۆبەیی پێشخراوو Priority Queue بەتاڵ بێت.

```
// get the head from the queue
  Integer head = prq.peek();
  System.out.println ( "Head of the queue is: "+ head);
```

ئەنجام//

Priority Queue = []

Priority Queue = [10, 15, 25, 20]

Head of the queue is: 10

رێگەی گەراندنەوە و لابردنی سەرە

E Poll()

ئەم رێگەیە بەکاردێت بۆ گەڕاندنەوە و لابردنی سەرە Head ((یەکەم دانەی پێکهێنەریی)) کیوو ی پێشخراوو Priority Queue ، هەروەها Null دەگێرێتەوە ئەگەر بەتاڵ Empty بێت.

```
// get the head from the queue
  Integer head = prq.poll();
  System.out.println ( "Head of the queue is: "+ head);
```

```
  System.out.println ( "Priority queue values after poll: "+ prq);
```

ئەنجام//

```
Priority Queue = []
Priority Queue = [10, 15, 25, 20]
Head of the queue is: 10
Priority queue values after poll: [15, 20, 25]
```

ڕێگەی گەڕاندنەوە و لابردنی سەرە

Boolean remove(Object o)

ئەم ڕێگەیە بەکاردەهێنرێت بۆ گەڕاندنەوە Retrieve و لابردنی Remove سەرە Head ((یەکەم دانە ی پێکهێنەر First Element)) ی کیووی پێشخراوو Priority Queue. هیچ داغڵکەرێك Parameter ی نییە.

```
    // remove 7 from the queue
  boolean isremoved = prq.remove(15);
  System.out.println ( "Return value after remove: "+ isremoved);
  System.out.println ( "Priority queue values after remove: "+ prq);
```

ئەنجام//

```
Priority Queue = []
Priority Queue = [10, 15, 25, 20]
Return value after remove: true
Priority queue values after remove: [10, 20, 25]
```

ڕێگەیی گەڕاندنەوەی نرخی ژمارەیی دانەکانی دیکیوو

Int size()

ئەم ڕێگەیە بەکاردێت بۆ زانینی حەجمی شارای نۆبەیی پێشخراوو Size of Priority Queue کە دەکاتە ژمارەی دانە پێکهێنەرەکانی ناوی. هیچ داغلّکەریکی Parameter نییە.

```
System.out.println ( "Size of the queue is: "+ prq.size());
```

ئەنجام//

```
Priority Queue = []
Priority Queue = [10, 15, 25, 20]
Size of the queue is: 4
```

لە بەرگرتنەوە بۆ ناو Array

Object[] toArray()

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی ئەرەیی Array کە هەموو دانەکانی شارای نۆبەیی پێشخراوو Priority Queue تیادا بێت.

```
// get objects from the queue
Object[] arr = prq.toArray();
System.out.println ( "Value in array: ");
for ( int i = 0; i<arr.length; i++ ){
System.out.println ( "Value: " + arr[i].toString()) ;
}
```

ئەنجام//

Priority Queue = []

Priority Queue = [10, 15, 25, 20]

Value in array:

Value: 10

Value: 15

Value: 25

Value: 20

لە بەرگرتنەوەی شارای نۆبەیی پێشخراوو بۆ ناو Array

<T> T[] toArray(T[]a)

ئەم ڕێگەیە بەکاردێت بۆ گەڕاندنەوەی ئەرەیی Array کە هەموو دانەکانی شارای نۆبەیی پێشخراوو Priority Queue تیابێت ، بەڵام حەجم دیاری دەکەین و، بەمەش کاتی لەبەر گرتنەوەکە یەکسانە بە ژمارەی دانەکان و دیاری کراوە و ورد و شیاوترە.

```
// create arr1
Integer[] arr1 = new Integer[5];
// use toArrsy() method
Integer[] arr2 = prq.toArray(arr1);
System.out.println ( "Value in arr1: ");

for ( int i = 0; i<arr1.length; i++ ){
System.out.println ( "Value: " + arr1[i]) ;
```

```
}
System.out.println ( "Value in arr2: ");
for ( int i = 0; i<arr2.length; i++ ){
System.out.println ( "Value: " + arr2[i]) ;
}
```

ئەنجام//

```
Priority Queue = []
Priority Queue = [10, 15, 25, 20]
Value in arr1:
Value: 10
Value: 15
Value: 25
Value: 20
Value: null
Value in arr2:
Value: 10
Value: 15
Value: 25
Value: 20
Value: null
```

شیکردنەوەی دەستەواژە

Expression Parsing

ڕێگەی نووسینی دەستەواژەی ژمێرەیی Arithmetic Expressionناسراوە بە نیشانەکان Notations، دەستەواژە ژمێرەییەکان Arithmetic Expression بە سێ شێوەی جیاواز دەنووسرێت بەڵام نیشانەکان هاوتا و یەکسانن لە هەر سێ شێوەکەدا، بەڵکو تەنها شوێنی نیشانەکان دەگۆڕێت، بە بێ ئەوەی بەرهەم و ئەنجامی Output دەربرینەکە بگۆڕێت.

نیشانەکان ئەمانەی لای خوارەوەن:

دانانی نیشانەکان لە نێوان دا Infix Notations

ئێمە لە شێوازی نووسین و دانانی نیشانەکان لە نێوان دا Infix Notation، نیشانەکان لەنێوان دانەکانی دەستەواژەکەدا دادەنێین و دەنووسین، وەکو :

A=b+c

لەم شێوەیەدا، نیشانەکان ((وەکو =،+)) لە نێوان دانەکانی دەستەواژەکەدا ((A,B,C)) دا بەکارهاتووە، بۆیە پێی دەڵێن ((دانانی نیشانەکان لە نێوان دا Infix Notations))، ئەمەش زۆر ئاسانە بۆ مرۆڤ، بۆ خوێندنەوە Read، نووسین Write و وتنی، بەڵام هەمان شت بۆ ئامێری کۆمپیوتەر گونجاو نییە و، شێوەکانی تر بەکاردەهێنین.

دانانی نیشانەکان لە پێشەوە Prefix Notations

لەم شێوەیەدا، نیشانەکان Operator لە پێشەوەی دانەکانی ناو دەستەواژەکەوە دادەنرێت و، دانە پێکهێنەرەکان بە دووایدا دادەنرێت:

+ab

لە نموونەکەدا هێمای (+) لە پێش (ab) دا نووسراوە، ئەمەش یەکسان و هاوتایە لەگەڵ دانانی نیشانەکان لە نێوان Infix Notations دانە پێکهێنەرەکانی بڕە Expression بیرکارییەکەدا، واتە a+b، دانانی نیشانەکان لە پێشەوە Prifix Notations بە نیشانەدانانی پۆڵەندی Polish Notation ناسراوە.

دانانی نیشانەکان لە دوواوە Postfix Notations

شێوازی ئەم نیشانەدانانە ناسراوە بە ((نیشانە دانانی پێچەوانەی پۆڵەندی Reversed Polish Notation)) ، لەم شێوەیەدا، نیشانەکان لە دوواوە ((پشتەوە)) دادەنرێت Postfix Notation، بۆ نموونە +ab، ئەمەش یەکسان و هاوتایە لەگەڵ دانانی نیشانەکان لە نێوان Infix Notations دانە پێکهێنەرەکانی بڕە Expression بیرکارییەکەدا، واتە a+b.

نموونەکان Examples

ئەم خشتەیەی خوارەوە، چەند نموونەیەکی تیادایە بۆ هەر سێ شێوەکەی نیشانە دانان، کە زیاتریش جیاوازی نێوانیان ڕوون دەکاتەوە:

S.n.	Infix Notation	Prefix Notation	Postfix Notation
1	a + b	+ a b	a b +
2	(a + b) * c	* + a b c	a b + c *
3	a * (b + c)	* a + b c	a b c + *
4	a / b + c / d	+ / a b / c d	a b / c d / +
5	(a + b) * (c + d)	* + a b + c d	a b + c d + *
6	((a + b) * c) - d	- * + a b c d	a b + c * d -

شیکردنەوەی دەستەواژەکان

Parsing Expressions

لە نەخشە سازیی Design ئەلگۆریسم Algorithm یان بەرنامە Programدا، شیکردنەوەی دانانی نیشانەکان لە نێوان دانەکان دا Infix Notation، پێویستە، چونکە یەکەمجار شێوەی دانانی نیشانەکان لە نێوان Infix Notation دانەکان دا، دەگۆڕین بۆ یەکێك لە دوو شێوەکەی تر، واتە بۆ دانانی نیشانەکان لە پێشەوە Prifix Notations یان دانانی نیشانەکان لە دوواوە Postfix Notations، پاشان کاری ژمێرەیی دەکرێت Arithmetic Operation.

بۆ شیکردنەوەی دەستەواژەی ژمێرەیی ((بیرکاری)) Arithmetic Expression پێویستە بەوردی ئاگاداری پێشرەویی precedence نیشانەکان و پەیوەندی نێوانیان بین.

a + b * c → a + (b * c)

کرداری لێکدان ((زەرب، کەرەت)) Multiplication لە پێشترە لە کرداری کۆکردنەوە ((کۆ، زائید)) Addition، واتە کرداری b*c لە پێشا ئەنجام دەدرێت، پاشان ئەی لەگەڵ ئەنجامی ئەم لێکدانە ((زەربە)) کۆدەکرێتەوە.

پەیوەندی پەیداکەر لە نێوانیان دا

Associativity

پەیوەندی پەیداکەر لە نێوانیان دا Associativity، وەسفی یاسایەك دەکات کاتێك دوو نیشانە هەمان پێشرەوییان هەبێت، لە دەستەواژەیەك دا، بۆ نموونە دەستەواژەی a+b-c، هەردوو نیشانەی + و -، هەمان پێشرەوییان Precedence هەیە، بەڵام پێویستە یەکێکیان لە پێشدا ئەنجام بدرێت، بۆیە پێویستمان بە ((پەیوەندی پەیداکەر لە نێوانیان دا Associativity)) هەیە لە نێوانیان دا، ئەویش چەپ بۆ راست ئەو پەیوەندی پەیدارەکەی نێوانیانە Left Associativity، بۆیە دەستەواژەکە بەم جۆرە ئەنجام دەدرێت: (a+b)-c.

ئەم خشتەیەی لای خوارەوە پێشرەوی Precedence و، پەیوەندی پەیداکە لە نێوانیان دا Associativity ڕوون دەکاتەوە:

S.n.	Operator	Precedence	Associativity
1	Esponentiation **^**	Highest	Right Associative
2	Multiplication (*****) & Division (**/**)	Second Highest	Left Associative
3	Addition (**+**) & Subtraction (**−**)	Lowest	Left Associative

لەم نموونەیەدا a+b*c، بەشێکی دەستەواژەکە b*c، یەکەم جار ئەنجام دەدرێت، چونکە پێشرەوویی بۆ کرداری لێکدانە، بەڵام ئەگەر ئێمە بمانەوێت لە پێشدا، a+b، ئەنجام بدرێت، ئەوا پێویستە سوود لە کەوانە وەربگرین و بەم جۆرە دەستەواژەکە بنووسین: (a+b)*c.

گەڕانەکان

Searches

گەڕانی هێڵی

Linear Search

گەڕانی هێڵی Linear Search، ئەلگۆریسمێکی زۆر سادەی گەرانە Search، لەم جۆرەی گەران دا، گەرانی زنجیرەی sequential search ڕوودەدات لەسەر هەموو دانە پێکهێنەرەکان Items یەك بە یەك، هەر یەکێك لە دانەکان دەپشکنێت و بەراوردی دەکات. ئەگەر دانەی لێکچوی دۆزییەوە ئەوا دەیگێڕیتەوە، بە پێچەوانەوە گەرانەکە بەردەوام دەبێت تا کۆتای زانیارییە کۆکراوەکە. بروانە نموونەکە:

-زانیاری کۆکراوەمان هەیە، وەك دەبینن:

10	14	19	26	27	31	33	35	42	44

-ئەگەر بمانەوێ بە شوێن ژمارە 33 دا بگەرێین، ئەوا لەگەڵ ژمارە 10 بەراوردی دەکات و، 10 یەکسان نییە بە 33 ، بۆیە لەگەڵ ژمارەی دوواتر بەراوردی دەکات و، بەو شێوەیە:

10	14	19	26	27	31	33	35	42	44
33									

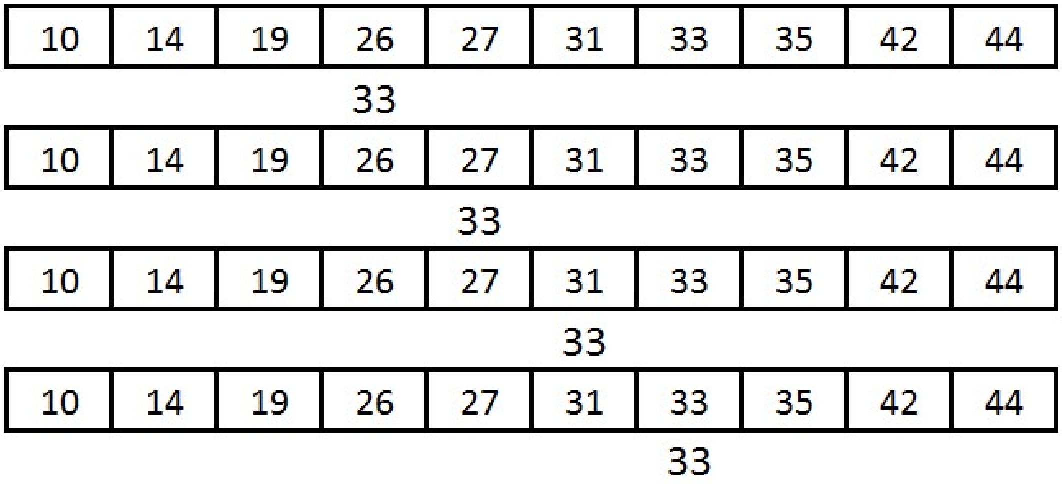
-بەهەمان شێوە بەراوردی زنجیرەی بەردەوام دەبێت، هەتا ژمارەکە دە دۆزیتەوە:

10	14	19	26	27	31	33	35	42	44
33									

10	14	19	26	27	31	33	35	42	44
	33								

10	14	19	26	27	31	33	35	42	44
		33							

-گەران بەردەوام دەبێت تا دەیدۆزیتەوە و، ئیتر دەوەستێت:



ئەلگۆریسمی گەرانی هێڵی

Linear Search Algorithm

ئەلگۆریسمی گەرانی هێڵی بەم جۆرەی لای خوارەوەیە:

```
Linear Search ( Array A, Value x)


Step 1: Set i to 1
Step 2: if i > n then go to step 7
Step 3: if A[i] = x then go to step 6
Step 4: Set i to i + 1
Step 5: Go to Step 2
Step 6: Print Element x Found at index i and go to
step 8
Step 7: Print element not found
Step 8: Exit
```

سویدۆکۆدی گەرانی هێڵی

Linear Search Pseudocode

سویدۆ کۆدی گەرانی هێڵی بەم شێوەیەی لای خوارەوەیە:

```
procedure linear_search (list, value)

   for each item in the list

      if match item == value

         return the item's location

      end if

   end for

end procedure
```

نموونەی جاڤا بۆ گەرانی هێلی

Java Example for Linear Search

```
/*
*Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
```

```
 */
package linear_search_db_book;
import java.util.Scanner;
/** @author HemnBarznji
 */
public class Linear_Search_DB_book {

   /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {


       int a[];
      Scanner s = new Scanner(System.in);
      System.out.println("Enter Size of an Array");
      int n = s.nextInt();
      a = new int[n];
      System.out.println("Enter " + n + " elements");
      for (int i = 0; i < n; i++) {
         a[i] = s.nextInt();
```

```
    }
    System.out.println("Enter Key element to Find");
    int key = s.nextInt();
    int position = linearSearch(a, key);
    if(position == -1){
        System.out.println("Element Not found");
    }else{
        System.out.println("Element Found at Position : "+position);
    }

}

public static int linearSearch(int a[], int key) {
    for (int i = 0; i < a.length; i++) {
        if (a[i] == key) {
            return i + 1;
        }
    }
    return -1;
}
public static void display(int a[]){
```

```
    System.out.println("Array Elements Are");
    for (int i = 0; i < a.length; i++) {
        System.out.println(a[i]+" ");


    }
  }
}
```

ئەنجام::

```
Enter Size of an Array
10
Enter 10 elements
10
14
19
26
27
31
33
35
42
44
```

Enter Key element to Find

33

Element Found at Position : 7

گەڕانی دووانی

Binary Search

گەڕانی دووانی Binary Search ئەلگۆریسمێکی خێرایی گەڕانە Fast Search Algorithm، کە ئالۆزێتی کاتی جێبەجێکردنی Run Time – Complexity بریتییە لە O(log n.)، ئەم ئەلگۆریسمی گەرانە کاردەکات لەسەر بنچینەیی ((دابەشکردن و زاڵبوون Divide and Conquer))، بۆ کار کردنی ئەم ئەلگۆریسمە بە شێوەیەکی درووست پێویستە زانیارییە کۆکراوەکە لە شێوەیەکی رێکخراوو Sorted Form دا.

گەرانی دووانی ((دوو دووی گەران)) Binary Search، گەرانە بە شوێن دانەیەکی دیاری کراوودا، بە بەکارهێنانی بەراورد کردنی ناوەراست Middle، ی کۆکراوەکە بەو دانەیەی دەمانەوێت.ئەگەر لێکچوون ڕووی دا، ئەوا ئیتدێکسی دانەکە دەگەرێنرێتەوە، ئەگەر دانەی ناوەراست Middle گەورەتر بوو لەو دانەیەی بە شوێنیا دەگەرێین، ئەوا گەران بۆ دانەکە دەکرێت لە نبمچە ریزیی Sub – Array لای راستی ناوەراستە Middle، بە پێچەوانەوە گەران بۆ دانەکەی دەمانەوێت دەکرێت لە نیکچە ریزی Sub – Array لای چەپ Left. ئەم کارە بەردەوام دەبێت لە سەر نیمچە ریزەکان Sub – Arrays هەتاوەکو حەجمی نیمچە ریز Sub – Array کەمدەبێتەوە بۆ سفر.

نموونەی گەرانی دووانی

Binary Search Example

ئەم نموونەیە ریزکراوەی رێکخراوو و ریزکراوە Sorted Array که بە بەکارهێنانی گەرانی دووانی ((دوو دوو گەران)) Binary Search، بە شوێن نرخی ((31)) دا دەگەرێین.

یەکەم جار، ناوەراستی Middle ڕیزەکە Array دیاری دەکەین، بە بەکارهێنانی ئەم هاوکێشە:

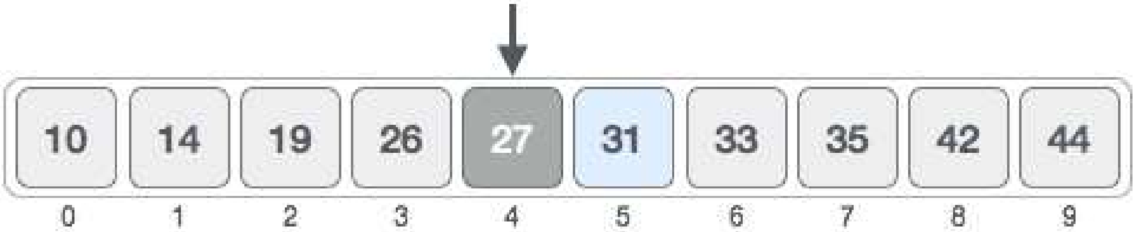
Mid = Low + (High – Low) / 2

نرخەکان دادەنێین بۆ دۆزینەوە ناوەراست:

Mid = 0 + (9-0) / 2

Mid=9/2=4.5

ژمارەی تەواوی 4.5 دەکاتە 4، بۆیە دانەی ناوەراست بریتییە لە 27:



دووای دیاری کردنی ناوەراست کە 27 بوو، ئەوا ئەوەی بە شوێنیا دەگەرێین کە 31 بەراورد دەکەین بە ناوەراست Middle و، 27 یەکسان نییە بە 31، لەبەر ئەوەی 31 کە بۆی دەگەرێین گەورەترە لە ناوەراست Middle کە 27 بوو، بۆیە لە نیمچە ریزیی لای راست Right Sub Arrayدا، بەشوێن نرخی 31 دا، دەگەرێین:

پاشان جارێکی تر، نزمترین و بەرزترین دیاری دەکەین و، بەبەکارهێنانی یاساکەی پێشوو، ناوەراست Middle دەدۆزینەوە:

low = mid + 1

mid = low + (high - low) / 2

دووای جێبەجێکردنی یاساکە:

Mid= 5 + (9-5)/2

Mid=5+(4)/2=5+2=7

ناوەراست، حەوتە، و ئەو نرخەی ئێمە بە دووایدا دەگەرێین ((31)) و یەکسان نییە بە ((35))، هەروەها بچوکترە لە ((35))، بۆیە نیمچە ریزیی لای چەپ Left Sub Array وەرەگرین،

جارێکی تر، ناوەراست ئەدۆزینەوە، کە دەکاتە 5، و نرخی ((31)) تیایە و، ئێمەش بە شوێن ((31)) دا دەگەرێین بۆیە کە بەراوردی دەکەین، دەبینین یەکسانە:

لە ئەنجامدا، نرخی 31 دەدۆزرێتەوە کە لە شوێنی 5 دایە.

سویدۆکۆد Pseudocode

سویدۆ کۆدی گەڕانی دووانی بەم جۆرەی لای خوارەوەیە:

```
Procedure binary_search
  A ← sorted array
  n ← size of array
  x ← value ot be searched
  Set lowerBound = 1
  Set upperBound = n
  while x not found
    if upperBound < lowerBound
      EXIT: x does not exists.

    set midPoint = lowerBound + ( upperBound - lowerBound ) / 2
    if A[midPoint] < x
      set lowerBound = midPoint + 1
    if A[midPoint] > x
      set upperBound = midPoint - 1
    if A[midPoint] = x
      EXIT: x found at location midPoint
  end while
end procedure
```

نموونەی جاڤا بۆ گەرانی دووانی

Java Example for Binary Search

```
/*
*Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
package binarysearch_implementation_db_book;

import java.util.Arrays;

/**@author HemnBarznji  */
public class BinarySearch_Implementation_DB_Book {
   public static void main(String[] args) {
    // initializing unsorted int array
  int intArr[] = {10,19,14,27,26,33,31,35,44,42};

  // sorting array
  Arrays.sort(intArr);

  // let us print all the elements available in list
```

```
  System.out.println("The sorted int array is:");
  for (int number : intArr) {
  System.out.println("Number = " + number);
  }
  // entering the value to be searched
  int searchVal = 31;
  int retVal = Arrays.binarySearch(intArr,searchVal);
  System.out.println("The index of element 31 is : " + retVal);
   }
}
```

گەرانی خستنە ناو

Interpolation Search

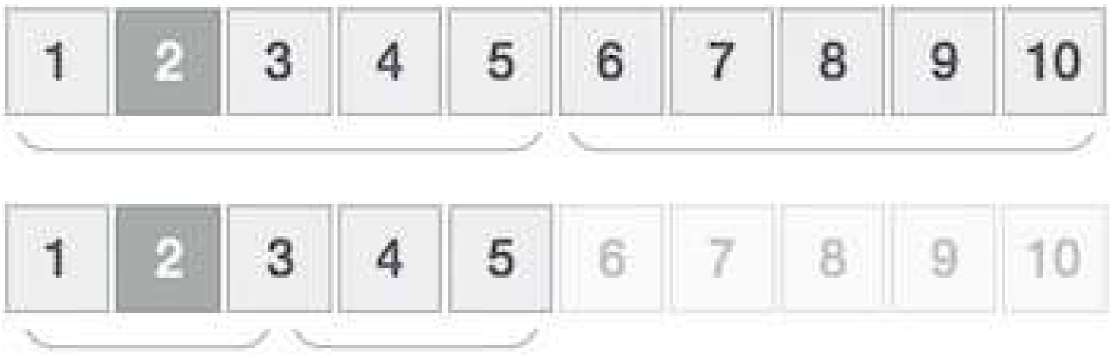
گەرانی خستنە ناو Interpolation Search جێگرەوەیەکی باشتر کراووی گەرانی دووانی Binary Search یە، ئەم ئەلگۆریسمی گەرانە کاردەکات لەسەر شوێنی پشکنینی Probing Position نرخی داواکراوو، ئەم ئەلگۆریسمە بەشێوەیەکی راست و درووست کاردەکات لەسەر زانیاری کۆکراوە بە شێوەی رێکخراوو و جیاکراوە Sort و دابەشکراوو بە شێوەیەکی یەکسان.

گەرانی دووانی Binary Search سوود و قازانجێکی زۆری هەیە لە ئاڵۆزێتی کاتدا Time Complexity لە گەرانی هێڵی Linear Searchدا، بەڵام ئاڵۆزێتی خراپترین حاڵەتی Worst – Case Complexity گەرانی هێڵی Linear Search بریتییە لە O(n)، لەکاتێکدا هی گەرانی دووانی Binary Search بریتییە لە O (log n).

دیاری کردنی شوێن لە گەرانی دووانی دا

Positioning in Binary Search

لە گەرانی دووانی Binary Searchدا، ئەگەر زانیاری داواکراوو و ویستراوو نەدۆزرێتەوە Not Found، پاشان لیستەکە دابەشبکرێت بۆ دوو (2) بەش، نزمتر Lower و بەرزتر Higher، پاشان گەران ئەنجام دەدرێت لە یەکێك لەم دوو بەشەدا.



له کاتێکدا، زانیارییەکە ڕێکخرا بێت، گەرانی دووانی سوودی نابێت بۆ گەران بە دووای شوێنە زانیاری داواکراوو.

شوێنی لێکۆڵینەوە لە گەرانی خستنە ناو دا

Position Probing in Interpolation Search

لە گەرانی خستنە ناو Interpolation Search، دا، گەران ئەنجام دەدرێت بە دووای دانەیەکی دیاری کراوو دا، بە ئەژمارکردنی شوێنی لێکۆڵینەوە ((شوێنی پشکنین)) Probe Position، لێکۆڵینەوە لە سەرەتای شوێن شوێنی گەورەی دانەکە ناوەڕاستی Middle کۆمەڵەکەیە.

ئەگەر لێکچوون ڕووی دا، پاشان ئیندێکسی Index دانەکە دەگەڕێنرێتەوە. بۆ دابەشکردنی لیستەکە بۆ دوو بەش ئەم ڕێگەیەی خوارەوە بەکاردەهێنین:

```
mid = Lo + ((Hi - Lo) / (A[Hi] - A[Lo])) * (X - A[Lo])

where −
   A    = list
   Lo   = Lowest index of the list
```

```
   Hi    = Highest index of the list
   A[n] = Value stored at index n in the list
```

ئەگەر ناوەراستی دانەکە گەورەتر بوو لە دانەکە ئەوا ئەوە شوێنی لێکۆلینەوەکە Probe Position دوو بارە حساباتی بۆ دەکرێتەوە لە نیمچە ریزیی Sub Array لای راستی ناوەراست Middleدا، بە پێچەوانەوە گەرانی بۆ ئەنجام دەدرێت لە نیمچە ریزیی Sub Array لای چەپ Left ی ناوەراست Middle دا، ئەم کردارە بەردەوام دەبێت هەتاوەکو حەجمی نیمچە ریز Size of Sub Array دەبێتە سفر.

ئالۆزێتی کاتی جێبەجێکردن Runtime Complexity ئەلگۆریسمی گەرانی خستنە ناو Interpolation Search Algorithm بریتییە لە O (log n (log n)) بە بەراورد بە O(log n) ی کاتی گەرانی دووانی Binary Search Time Complexity لە باشترین حالّەت دا.

ئەلگۆریسمی گەرانی خستنە ناو

Interpolation Search Algorithm

Step 1 − Start searching data from middle of the list.

Step 2 − If it is a match, return the index of the item, and exit.

Step 3 − If it is not a match, probe position.

Step 4 − Divide the list using probing formula and find the new midle.

Step 5 − If data is greater than middle, search in higher sub-list.

Step 6 − If data is smaller than middle, search in lower sub-list.

Step 7 − Repeat until match.

سویدۆ کۆدی گەرانی خستنە ناو

Interpolation Search Pseudocode

```
A → Array list

N → Size of A

X → Target Value


Procedure Interpolation_Search()


  Set Lo  →  0

  Set Mid → -1

  Set Hi  →  N-1


  While X does not match

    if Lo equals to Hi OR A[Lo] equals to A[Hi]

      EXIT: Failure, Target not found

    end if

    Set Mid = Lo + ((Hi - Lo) / (A[Hi] - A[Lo])) * (X - A[Lo])

    if A[Mid] = X

      EXIT: Success, Target found at Mid

    else

      if A[Mid] < X
```

```
        Set Lo to Mid+1
      else if A[Mid] > X
        Set Hi to Mid-1
      end if
    end if
  End While
End Procedure
```

نموونەی جاڤا بۆ گەرانی خستنە ناو

Java Example for Interpolation Search

```
/*
 *Data Structure Book -  Kurdish Language
 * Written By: Hemn Mala Karim Barznji
 * Software Development Instructor
 */
package interpolationsearch_implementation_db_book;
import java.util.*;
/**@author HemnBarznji */
public class InterpolationSearch_Implementation_DB_Book {

   /** interpolationSearch function **/
   public static int interpolationSearch(int[] sortedArray, int toFind)
```

```
{
    int low = 0;
    int high = sortedArray.length - 1;
    int mid;
    while (sortedArray[low] <= toFind && sortedArray[high] >=
toFind)
    {   if (sortedArray[high] - sortedArray[low] == 0)
            return (low + high)/2;
        /** out of range is possible  here **/
         mid = low + ((toFind - sortedArray[low]) * (high - low)) /
(sortedArray[high] - sortedArray[low]);
         if (sortedArray[mid] < toFind)
            low = mid + 1;
         else if (sortedArray[mid] > toFind)
            high = mid - 1;
         else
            return mid;     }
    if (sortedArray[low] == toFind)
        return low;
       /** not found **/
    else
        return -1;   }
```

```
/** Main method **/
public static void main(String[] args)
{   Scanner scan = new Scanner( System.in );
    System.out.println("Interpolation Search Test\n");
    int n, i;
    /** Accept number of elements **/
    System.out.println("Enter number of integer elements");
    n = scan.nextInt();
    /** Create integer array on n elements **/
    int arr[] = new int[ n ];
    /** Accept elements **/
    System.out.println("\nEnter "+ n +" sorted integer elements");
    for (i = 0; i < n; i++)
        arr[i] = scan.nextInt();
    System.out.println("\nEnter element to search for : ");
    int key = scan.nextInt();

    int result = interpolationSearch(arr, key);
    if (result == -1)
        System.out.println("\n"+ key +" element not found");
    else
```

```
        System.out.println("\n"+ key +" elemnt found at position "+
result);    }        }
```

ئەنجام::

```
Interpolation Search Test

Enter number of integer elements

10

Enter 10 sorted integer elements

1

2

3

4

5

6

7

8

9

10


Enter element to search for :

2

2 elemnt found at position 1
```

گەڕانی خشتەی بەش بەش کردن

Hash Table Search

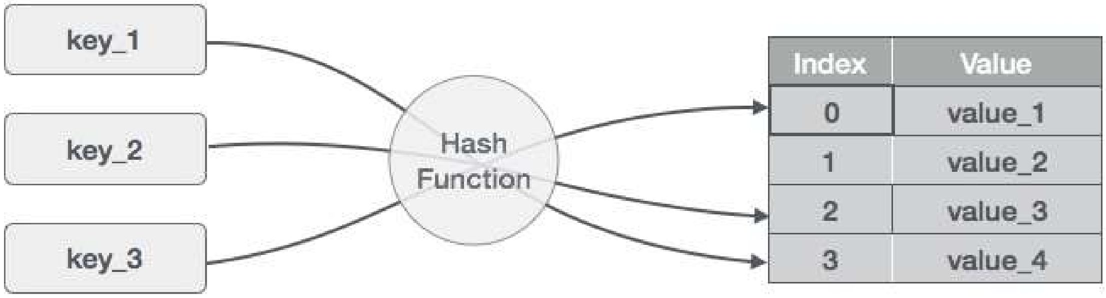
خشتەی بەش بەشکردن ((خشتەی لەت لەتکردن)) Hash Table شێوازێکی پێکهاتنی زانیارییە کە زانیاری پاشکەوت دەکات لە شێوەی ئەنجومەنی پەیوەست Associative Manner، لە خشتەی بەش بەشکراوو Hash Tableدا، زانیاری پاشەکەوت دەکرێت لە شێوەی ریز Array Formدا، کە نرخی هەر زانیارییەك Data Value یەك نرخی شوێنی Index Value هەیە و، تەنهایە ئەم ئیندێکسە، دەستگەیشتن بەم زانیارییە زۆر خێرایە ئەگەر ئیندێکسی Index زانیاری داواکراو و پێویست بزانرێت.

هەروەها کرداری دانان ((خستنە ناو Insert)) و گەران ((Search)) زۆر خێرایە و، ئەم خشتەی بەش بەشکردنە ریز Array بەکاردەهێنێت وەکو ناوەندی شارا ((کۆگا، مەخزەن Storage)) و، شێوازی بەشبەشکردنیش Hash Technique بەکاردەهێنێت بۆ درووستکردنی ئیندێکس Index کە زانیاری تیادادەنرێت یان زانیاری تێداهەیە.

بەشبەشکردن

Hashing

بەشبەشکردن ((لەتلەتکردن Hashing)) شێوازی گۆرینی نرخی کۆمەڵەی کلیلەکان بۆ کۆمەڵەی ئیندێکسەکانی ریز Indexes of Array، کە کرداری مۆدیۆلۆ ((ماوە Modulo)) بەکاردەهێنین بۆ کۆمەڵەی نرخەکانی کلیل، بۆ نموونە خشتەیەکی بەش بەشکردن حەجمەکەی Size بیستە 20، ئەم زانیارییانەی لای خوارەوە پاشکەوت کراوە تیایدا، شێوەی دانە پێکهێنەرەکان (کلیل،نرخ) (Key,Value).



- (1,20)
- (2,70)
- (42,80)
- (4,25)
- (12,44)
- (14,32)
- (17,11)
- (13,78)
- (37,98)

S.n.	Key	Hash	Array Index
1	1	1 % 20 = 1	1
2	2	2 % 20 = 2	2
3	42	42 % 20 = 2	2
4	4	4 % 20 = 4	4
5	12	12 % 20 = 12	12

6	14	14 % 20 = 14	14
7	17	17 % 20 = 17	17
8	13	13 % 20 = 13	13
9	37	37 % 20 = 17	17

پشکنینی هێڵی

Linear Probing

لە وانەیە پشکنینی ((لێکۆلینەوەی)) هێڵی Linear Probing ڕووبدات، کە شێوازی بەشبەشکردن Hash Technique بەکارهاتبێت بۆ درووستکردنی ئیندیکسی Index ریزەکە Array، لەم حاڵەتەدا، دەگەرێین بۆ شوێنی بەتاڵی دوواتر لە ریزەکە Arrayدا، بە ڕووانین بۆ خانەکەی دوواتر تا دۆزینەوەی خانەی بەتاڵ Found an Empty Cell، ئەم تەکنیکە پێی دەڵێن Linear Probing.

S.n.	Key	Hash	Array Index	After Linear Probing, Array Index
1	1	1 % 20 = 1	1	1
2	2	2 % 20 = 2	2	2
3	42	42 % 20 = 2	2	3

4	4	4 % 20 = 4	4	4
5	12	12 % 20 = 12	12	12
6	14	14 % 20 = 14	14	14
7	17	17 % 20 = 17	17	17
8	13	13 % 20 = 13	13	13
9	37	37 % 20 = 17	17	18

کرداره بنچینەییەکان

Basic Operations

کرداره بنچینەییەکانی گەرانی خشتەی بەش بەشکردن Hash Table Search ئەمانەی لای خوارەوەن، کە بە کورتی باسیا دەکەین:

کرداری گەران

Search Operation

```
struct DataItem *search(int key){
   //get the hash
   int hashIndex = hashCode(key);
   //move in array until an empty
   while(hashArray[hashIndex] != NULL){
```

```
    if(hashArray[hashIndex]->key == key)
      return hashArray[hashIndex];
    //go to next cell
    ++hashIndex;
    //wrap around the table
    hashIndex %= SIZE;
  }
  return NULL;
}
```

کرداری دانان

Insert Operation

```
void insert(int key,int data){
  struct DataItem *item = (struct DataItem*) malloc(sizeof(struct
DataItem));
  item->data = data;
  item->key = key;
  //get the hash
  int hashIndex = hashCode(key);
  //move in array until an empty or deleted cell
  while(hashArray[hashIndex] != NULL && hashArray[hashIndex]-
>key != -1){
```

```
    //go to next cell
    ++hashIndex;
    //wrap around the table
    hashIndex %= SIZE;
  }
  hashArray[hashIndex] = item;
}
```

کرداری سرینەوە

Delete Operation

```
struct DataItem* delete(struct DataItem* item){
  int key = item->key;
  //get the hash
  int hashIndex = hashCode(key);
  //move in array until an empty
  while(hashArray[hashIndex] !=NULL){
    if(hashArray[hashIndex]->key == key){
      struct DataItem* temp = hashArray[hashIndex];

      //assign a dummy item at deleted position
      hashArray[hashIndex] = dummyItem;
      return temp;
```

```
    }
    //go to next cell
    ++hashIndex;
    //wrap around the table
    hashIndex %= SIZE;
  }
  return NULL;
}
```

پۆلی خشتەی بەشبەشکردن

Hash Table Class

لە زمانی بەرنامە سازیی جاڤا دا، پۆلێکی ئامادەکراوو بەردەستە بۆ جێبەجێکردنی خشتەی بەش بەشکردن، کە لە چەند درووست کەرێك Constructors و، چەند رێگەیەك پێکهاتووە، کە بە کورتی باسیان دەکەین، لەگەڵ نموونەی کۆد بۆ رێگەکان:

دروستکەرەکان

Constructors

لەم خشتەیەدا، دروستکەرەکانمان ڕوون کردۆتەوە:

Sr.No	دروستکەر و وەسف Constructor and Description
1	Hashtable( )

	دروستکەری هەمیشەیی خشتەی بەشبەشکردنە.
2	**Hashtable(int size)** **ئەم دروستکەرە داغڵکەری ژمارەی تەواو قبوڵدەکات، و هەروەها دروستکردنی خشتەی بەش بەشکردن کە حەجمەکەی دیاری کرابێت بە نرخێکی ژمارەی تەواو.**
3	**Hashtable(int size, float fillRatio)** **هەروەها دروستکردنی خشتەی بەش بەشکردن کە حەجمەکەی دیاری کرابێت بە نرخێکی ژمارەی تەواو و ڕێژەیی پڕ Fill Ratio بە بەکارهێنانی fiiRatio، ئەم ڕێژەیە پێویستە لە نێوان 0.0بۆ 1.0 دا بێت.**

Methods ڕێگەکان

ژمارە	ڕێگەکان و بەکاهێنانیان
1	**void clear( )** **بۆ سرینەوە و بەتاڵکردنەوەی خشتەی بەش بەشکردن بەکاردێت.**
2	**Object clone( )** **گەراندنەوەی دوو هێندە کردن و لەبەر گرتنەوەی دانەکان.**
3	**boolean contains(Object value)** **گەراندنەوەی نرخی ڕاست True، ئەگەر نرخەکە یەکسان بوو بە نرخی خشتەی بەش بەشکردنەکە. گەراندنەوەی نرخی هەڵە False ئەگەر نرخەکە نەدۆزرایەوە.**
4	**boolean containsKey(Object key)** **گەراندنەوەی نرخی ڕاست True، ئەگەر کلیل Key یەکسان بوو بە کلیلی ناو خشتەی بەشبەشکردن Hash Table، گەراندنەوەی نرخی هەڵە ئەگەر کلیلەکەی**

	نەدۆزییەوە.
5	**boolean containsValue(Object value)** گەراندنەوەی نرخی راست **True**، ئەگەر نرخەکە یەکسان بوو بە نرخی خشتەی بەش بەشکردنەکە. گەراندنەوەی نرخی هەڵە **False** ئەگەر نرخەکە نەدۆزرایەوە.
6	**Enumeration elements( )** گەراندنەوەی ژمارە دانەی نرخەکان کە لە خشتەی بەش بەشکردن دایە.
7	**Object get(Object key)** گەراندنەوەی تەنەکە **The Object** کە نرخی پەیوەند بە کلیلی تیابێت، ئەگەر کلیلەکە لە خشتەی بەشبەشکردن **Hash Table**دا نەبوو، ئەوا تەنی هیچ شتێك ((بەتاڵ)) **Null Object** دەگێریتەوە.
8	**boolean isEmpty( )** گەراندنەوەی راست **True** ئەگەر خشتەی بەشبەشکردن **Hash Table** بەتاڵبوو **Empty**، بەڵام هەڵە **False** دەگێریتەوە ئەگەر لانی کەم کلیلێکی تیابوو.
9	**Enumeration keys( )** گەراندنەوەی ژمارە دانەی کلیلەکان کە لەناو خشتەی بەشبەشکردنەکە دایە.
10	**Object put(Object key, Object value)** داغڵکردن و دانانی کلیل **Key** و نرخ **Value** بۆ ناو خشتەی بەشبەشکردن **Hash Table**،
11	**void rehash( )** زیاد کردنی حەجمی خشتەی بەشبەشکردن و دووبارە بەشکردنەوەی هەموو کلیلەکان.
12	**Object remove(Object key)** لابردنی کلیل **Key** و نرخەکەی، گەراندنەوەی نرخی پەیوەند بە کلیلەکەوە، بەڵام ئەگەر کلیلەکە نەبوو ئەوا تەنی هیچ شتێك ((بەتاڵ)) **Null Object** دەگێرێتەوە.
13	**int size( )** گەراندنەوە ژمارەی ناو خشتەی بەش بەشکردنەکە، واتە حەجمەکەی.

14	**String toString( )** گەراندنەوەی سترینگەکە کە هاوتایی خشتەی بەشبەشکردن بێت.

Example نموونە

```
import java.util.*;
public class HashTableDemo {
  public static void main(String args[]) {
    // Create a hash map
    Hashtable balance = new Hashtable();
    Enumeration names;
    String str;
    double bal;

    balance.put("Zara", new Double(3434.34));
    balance.put("Mahnaz", new Double(123.22));
    balance.put("Ayan", new Double(1378.00));
    balance.put("Daisy", new Double(99.22));
    balance.put("Qadir", new Double(-19.08));

    // Show all balances in hash table.
    names = balance.keys();
    while(names.hasMoreElements()) {
      str = (String) names.nextElement();
```

```
      System.out.println(str + ": " +
      balance.get(str));
    }
    System.out.println();
    // Deposit 1,000 into Zara's account
    bal = ((Double)balance.get("Zara")).doubleValue();
    balance.put("Zara", new Double(bal+1000));
    System.out.println("Zara's new balance: " +
    balance.get("Zara"));
  }
}
```

ئەنجام::

```
Qadir: -19.08
Zara: 3434.34
ahnaz: 123.22
Daisy: 99.22
Ayan: 1378.0
Zara's new balance: 4434.34
```

وشەکان و واتاکان

Array n.

ڕیز. کۆمەڵێك دانە لە شێوەی ریز دا. یەك بە دووای یەك.

Array v.

ڕیز دەگرێ.

Link n.

عەلقە. بەستەر. برگە. بەش. بەند. پەیوەندی. پێکەوە گرێدەر.

Link v.

پێکەوە دەبەستێ . گرێ دەدا. پێکەوە بەستن. لە یەکدان.

Implementation n.

ئیشپێکردن. پەیرەوکردن. جێبەجێکردن. بەجێهێنان.

Application n.

بەکارهێنان. کارپێکردن. شێوەی بەکارهێنان.

Pop v.

لە پڕ سەر دەرهێنان. دەرهێنانەوە. گەراندنەوەی بە پەلە. هەڵدان.

Stack n.

شارا. ڕیزە ڕەفە، کۆمەڵە.

Stack v.

کەڵەکە کردن. هەڵچنین. بار کردن، لێ بار کردن.

Push v.

پاڵپێوەنان، پاڵدان. پەستاوتن. پاڵنان. خستنە ناو.

Push n.

پاڵ. پەستان. دەستە. کۆمەڵ. تەکان.

Abstract

پوختە. مانایی. رووت.

Queue n.

ڕیز. ڕیزە زانیاریی. نۆرە. نۆبە. نۆبەیی.

Queue v.

ریز دەگرێ. سەرە دەگرێ. نۆبە دەگرێ.

Rear n.

پشتەوە. دوواوە. دووا.

Front n.

پێشەوە. پێش. ڕوو. بەر دەم.

Priority n.

پێشتری. بەرتری. گرنگتری. مافی لە پێشتربوون. مافی پێشخستن. شتێکی پێشخراوو.

Exception n.

بێجگە. نا ئاسا. بێجگە کاری. لێ دەرکردن. جیاکاری. پەسەند نەکردن.

Poll n.

تەوقە سەر. سەر.

Remove n.

لابردن. جوڵان. گوستنەوە. گوێزانەوە.

Remove v.

لادەبات. دەگویزێتەوە. لێ لابردن. لێ جیاکردنەوە.

Offer v.

نرخ پێدان. پێبەخشین. بۆ کردن. پێشکەشکردن.

Get v.

چنگ کەوتن. دەستگیربوون. کردن. بردن. بوون.

Set v.

دانان. خستنە سەر. لە سەردانان. سازاندن. ڕێکخستن. تیادان. تێخستن.

Expression n.

دەربڕین. نیشانەی جەبری یان بیرکاری. دەستەواژە.

Parse v.

شیکرانەوەی ڕستە. شیکردنەوەی ڕێزمانی. شیکردنەوە.

Infix v.

تێخستن و تیادا چەسپکردن. تێدا قایم کردن. ناوگر. ناوبەند.

Notation n.

تێبینی. تێبینیکاری. کۆمەڵێك نیشانە. یاد داشت.

Prefix n.

پێشگر. پێشبەند. پێشکۆ. دانان لە پێشەوە.

Postfix n.

پاشگر. پاشگر پێوەنان.

Conquer v.

دەست بە سەرا گرتن. شکاندن. داگیر کردن. زاڵبوون.

Sort n.

جۆر. چەشن. شێواز. رێ.

Sort v.

هاوپۆلکردن. جیا کردنەوە. بابەت بابەت کردن.