CS2113 Lab: Collections

Yawei Wang

10/29/2018

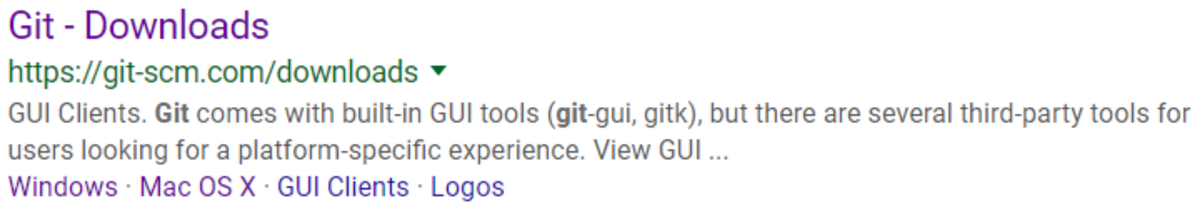
Install and Use IntelliJ on Mac or Window

- If you haven't installed JDK before, go to https://www.oracle.com/technetwork/java/javaseproducts/downloads/index.html
- Google IntelliJ and click on download link.
- Download the community version and install it.
- Create a new project.
- Choose project JDK: New… -> JDK -> (Your system should automatically choose the correct folder) -> OK
- Set the project location.
- Create a "Hello, World!" program and run it.

For Windows users, if your system doesn't choose the correct folder, manually select C:\Program Files\Java\jdk1.8.0

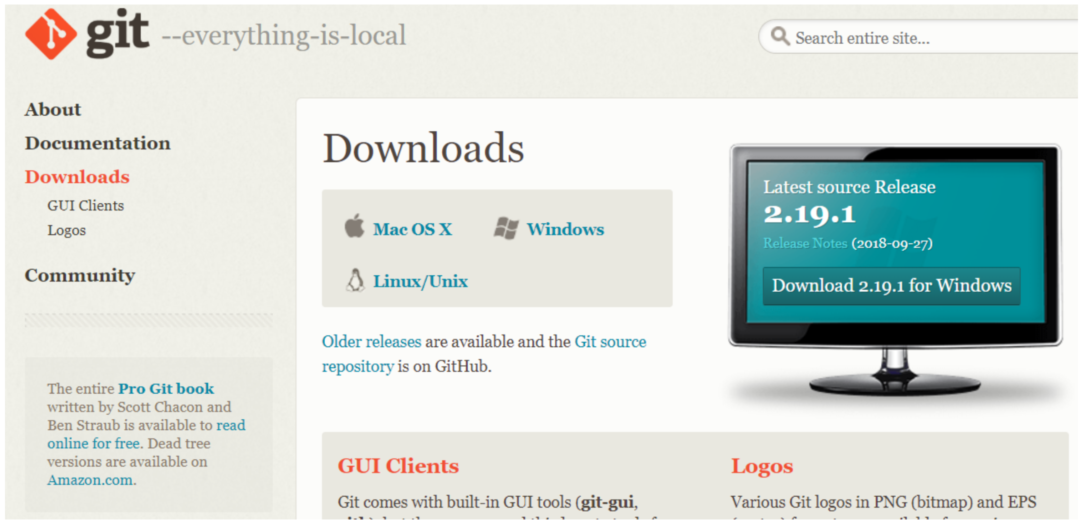
Install Git

- Google “git download” and click on



Git - Downloads
https://git-scm.com/downloads
GUI Clients. **Git** comes with built-in GUI tools (**git**-gui, gitk), but there are several third-party tools for users looking for a platform-specific experience. View GUI ...
Windows · Mac OS X · GUI Clients · Logos

- Download the correct version of git based on your operating system and install it.



git --everything-is-local

Search entire site...

About
Documentation
Downloads
GUI Clients
Logos
Community

The entire **Pro Git book** written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

Downloads

Mac OS X
Windows
Linux/Unix

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.19.1
Release Notes (2018-09-27)
Download 2.19.1 for Windows

GUI Clients
Git comes with built-in GUI tools (**git-gui**,

Logos
Various Git logos in PNG (bitmap) and EPS

Checkout a Project from GitHub

- Register an existing GitHub account.
  - Windows: **File | Settings | Version Control | GitHub | Click "+".**
  - Mac: **Preferences | Version Control | GitHub | Click "+".**
- Clone a repository
  - From the top main menu, choose **VCS | Checkout from Version Control | Git**.
  - In the **Clone Repository** dialog that opens, specify the URL of the repository that you want to clone.
  - In the **Directory** field, specify the path where the folder for your local Git repository will be created into which the remote repository will be cloned.
  - Click **Clone**. If you want to create a IntelliJ IDEA project based on the sources you have cloned, click Yes in the confirmation dialog. Git root mapping will be automatically set to the project root directory.

Troubleshooting: For Windows users, if IntelliJ can't locate your git, please set the path manually.
File -> Settings -> Version Control -> Git -> Path to Git executable
Change the path to C:\Program Files\Git\cmd\git.exe

Commit and Push Changes

Commit changes locally

- To invoke the Commit Changes dialog, select the files (or an entire changelist) in the Local Changes view and click ✓ icons on the toolbar or choose **Commit Changes** on the context menu of the selection; or press **Ctrl+K.**
- Enter a **commit message** and select the Before Commit actions you want IntelliJ IDEA to perform before committing the selected files to the local repository.
- Click the Commit button.

Push changes to a remote repository

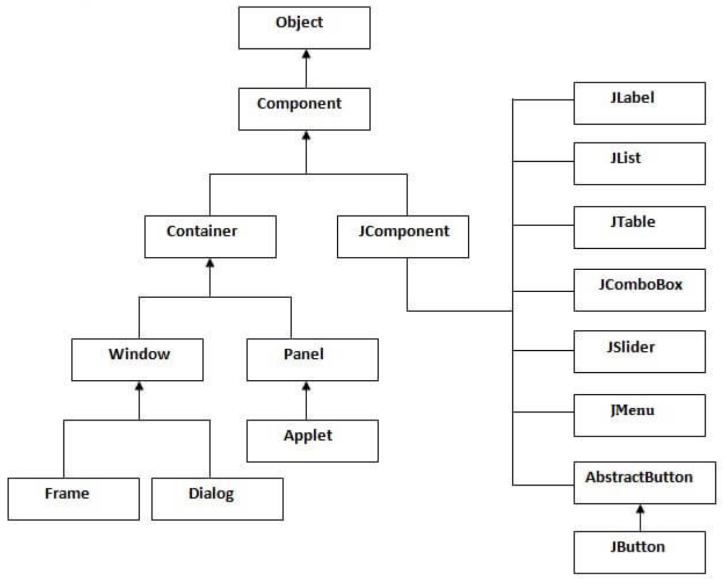
- Press **Ctrl+Shift+K** or choose **VCS | Git | Push** from the main menu.
- Click the **Push** button when ready.

What is a framework in Java?

- It provides readymade architecture.
- It represents a set of interfaces and classes (APIs).
- It is optional.

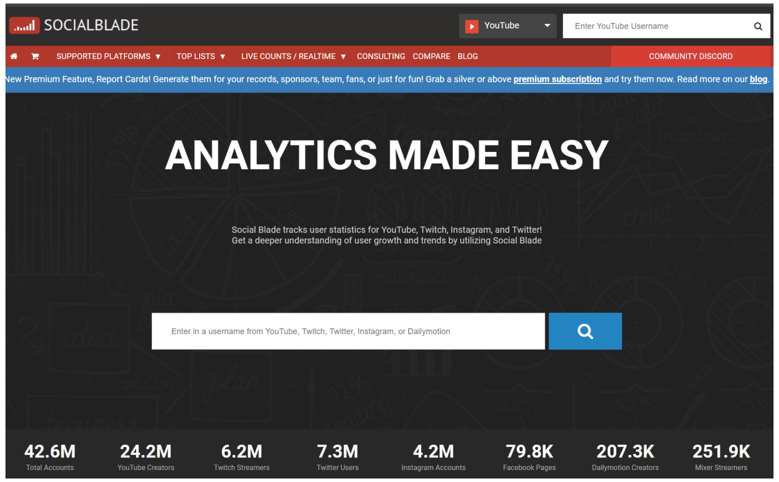
Example:

- Java Swing framework (GUI framework)
- Java Collection framework



Hierarchy of Java Swing classes

What is an API (Application Program Interface)?



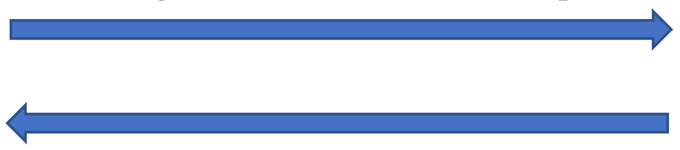
New Premium Feature, Report Cards! Generate them for your records, sponsors, team, fans, or just for fun! Grab a silver or above **premium subscription** and try them now. Read more on our **blog**.

ANALYTICS MADE EASY

Social Blade tracks user statistics for YouTube, Twitch, Instagram, and Twitter!
Get a deeper understanding of user growth and trends by utilizing Social Blade

42.6M	24.2M	6.2M	7.3M	4.2M	79.8K	207.3K	251.9K
Total Accounts	YouTube Creators	Twitch Streamers	Twitter Users	Instagram Accounts	Facebook Pages	Dailymotion Creators	Mixer Streamers

SocialBlade can help users track YouTube Channel Statistics, Twitch User Stats, Instagram Stats, and much more.


Using the YouTube API to send requests
Sending the data back to the client

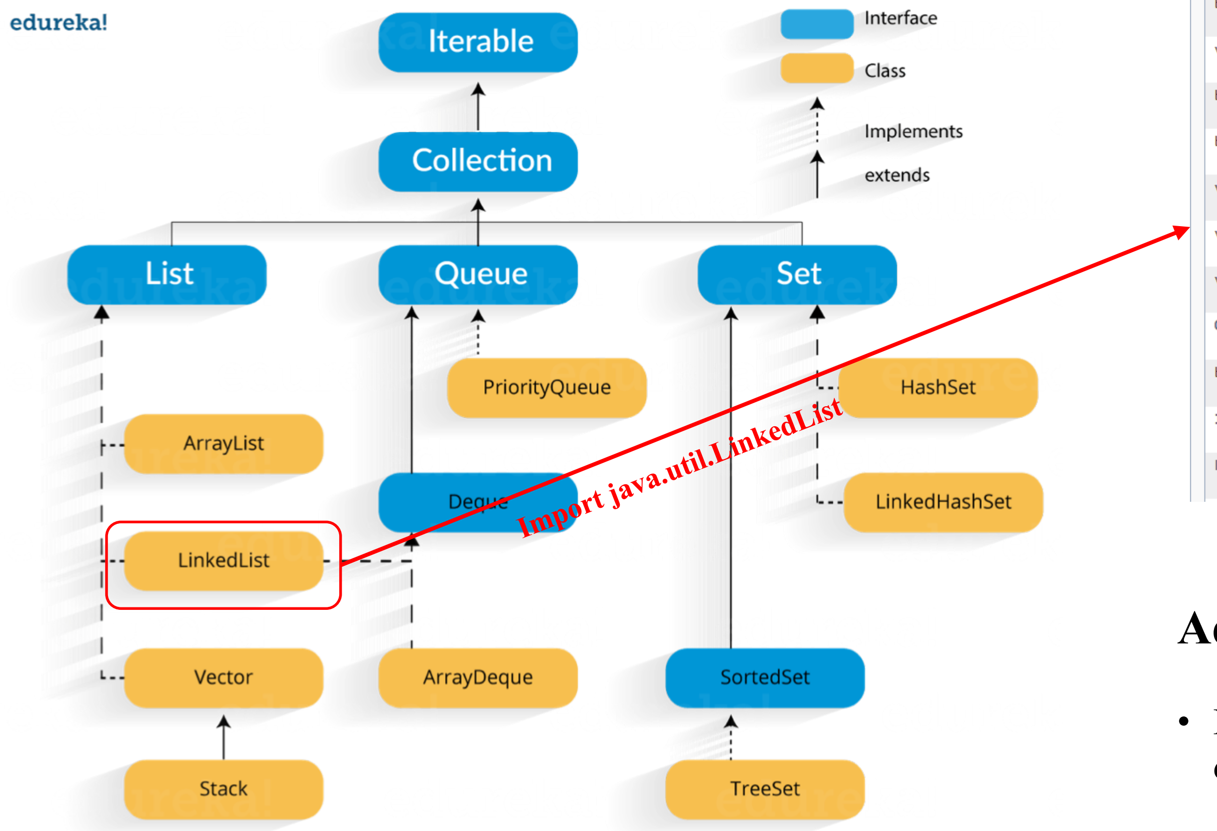
YouTube API allows developers to access video statistics and YouTube channels' data.

- API is a software intermediary that allows two applications to talk to each other.
- Java API is a list of all classes that are part of the Java development kit (JDK). It includes all Java packages, classes, and interfaces, along with their methods, fields, and constructors.

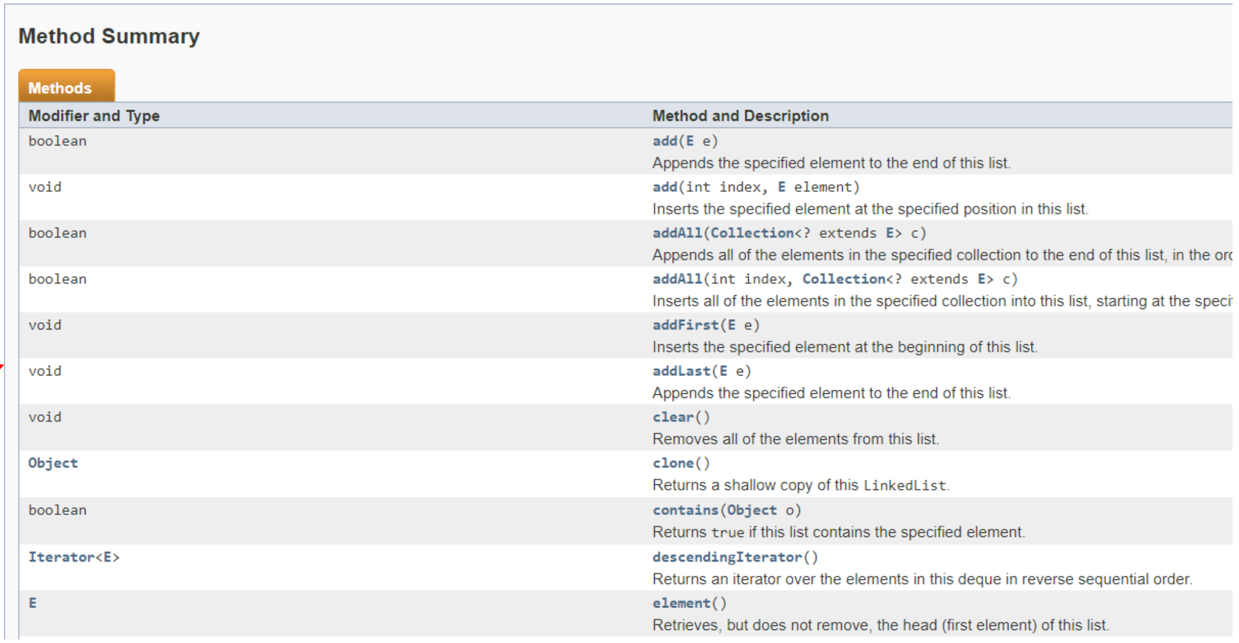
Collections in Java

- The **Collection in Java** is a framework that provides an architecture to store and manipulate the group of objects.
- All the operations that you perform on a data such as searching, sorting, insertion, manipulation, deletion, etc. can be achieved by Java Collections.
- Java Collection means a single unit of objects. Java Collection framework provides many interfaces (Set, List, Queue, Deque, etc.) and classes (ArrayList, Vector, LinkedList, PriorityQueue, HashSet, LinkedHashSet, TreeSet, etc.).

Java Collection Framework



Java Collection Framework Hierarchy



Method Summary

Methods

Modifier and Type	Method and Description
`boolean`	`add(E e)` Appends the specified element to the end of this list.
`void`	`add(int index, E element)` Inserts the specified element at the specified position in this list.
`boolean`	`addAll(Collection<? extends E> c)` Appends all of the elements in the specified collection to the end of this list, in the or
`boolean`	`addAll(int index, Collection<? extends E> c)` Inserts all of the elements in the specified collection into this list, starting at the speci
`void`	`addFirst(E e)` Inserts the specified element at the beginning of this list.
`void`	`addLast(E e)` Appends the specified element to the end of this list.
`void`	`clear()` Removes all of the elements from this list.
`Object`	`clone()` Returns a shallow copy of this `LinkedList`.
`boolean`	`contains(Object o)` Returns `true` if this list contains the specified element.
`Iterator<E>`	`descendingIterator()` Returns an iterator over the elements in this deque in reverse sequential order.
`E`	`element()` Retrieves, but does not remove, the head (first element) of this list.

Advantages of Collection Framework:

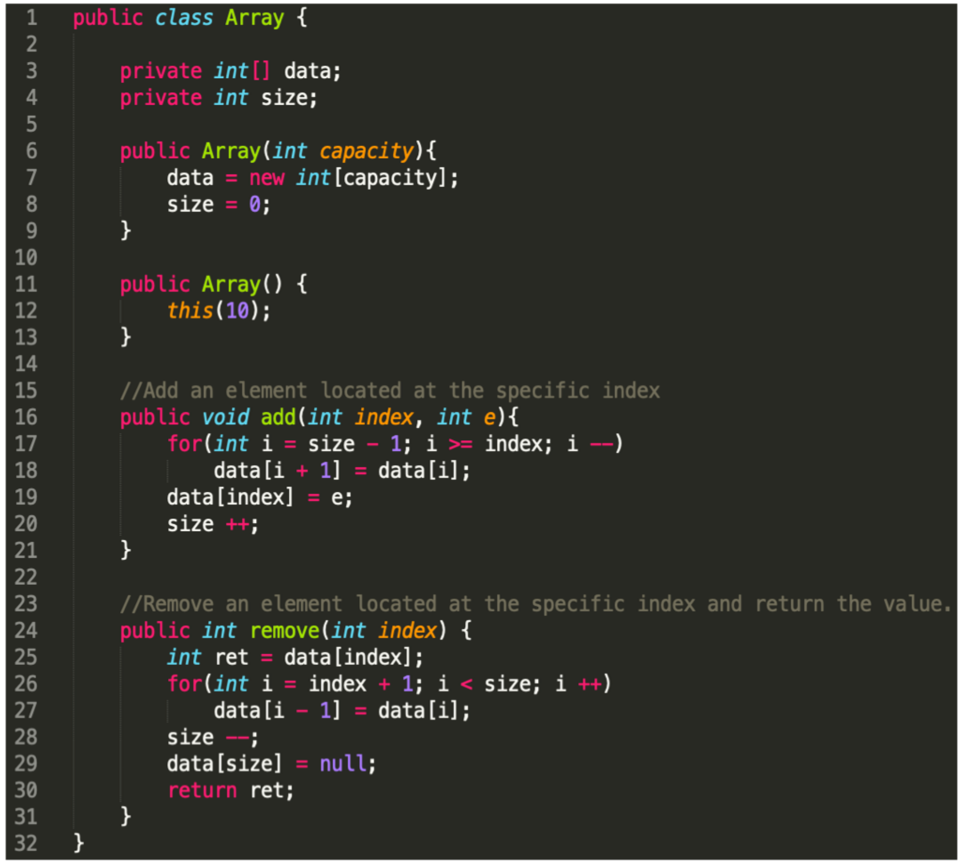
- **Reduces programming effort**: A programmer doesn’t have to worry about the design of Collection, and he can focus on its best use in his program.
- **Increases program speed and quality**: Increases performance by providing high-performance implementations of useful data structures and algorithms.

Generics in Java 1

Generics enable *types* (classes and interfaces) to be parameters when defining classes, interfaces and methods. **Generic types are extensively used in Java collections**.

Why use generics?

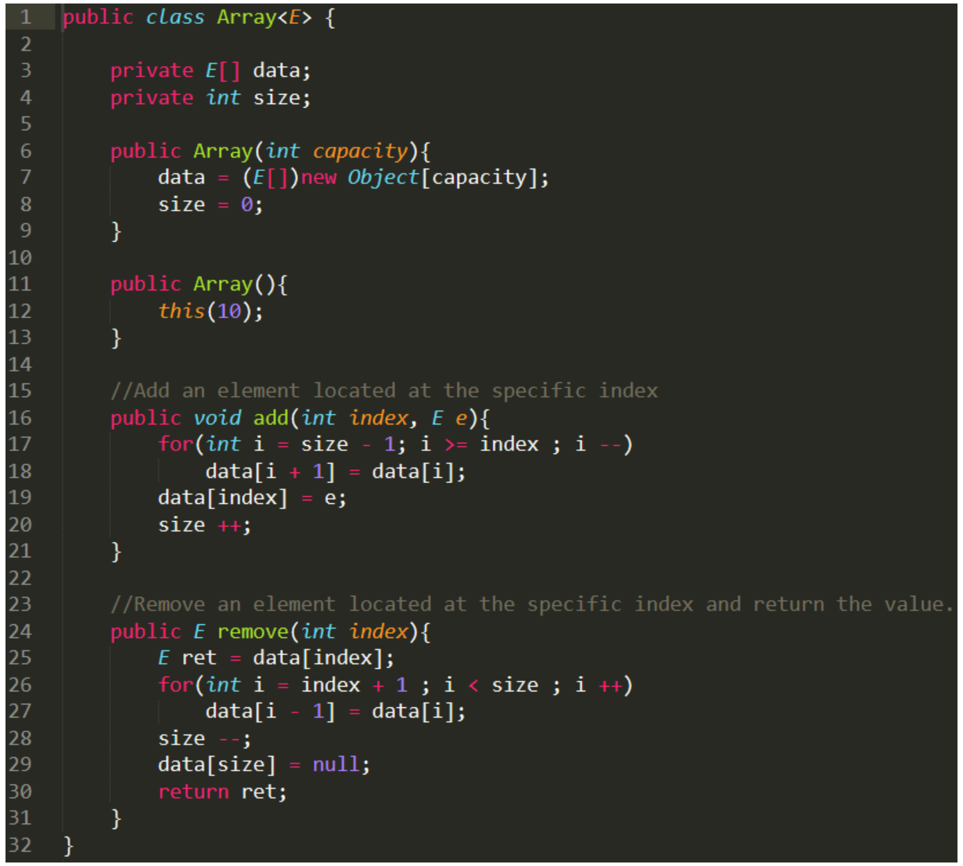
non-generic code



```
public class Array {

    private int[] data;
    private int size;

    public Array(int capacity){
        data = new int[capacity];
        size = 0;
    }

    public Array() {
        this(10);
    }

    //Add an element located at the specific index
    public void add(int index, int e){
        for(int i = size - 1; i >= index; i --)
            data[i + 1] = data[i];
        data[index] = e;
        size ++;
    }

    //Remove an element located at the specific index and return the value.
    public int remove(int index) {
        int ret = data[index];
        for(int i = index + 1; i < size; i ++)
            data[i - 1] = data[i];
        size --;
        data[size] = null;
        return ret;
    }
}
```

VS.

generic code



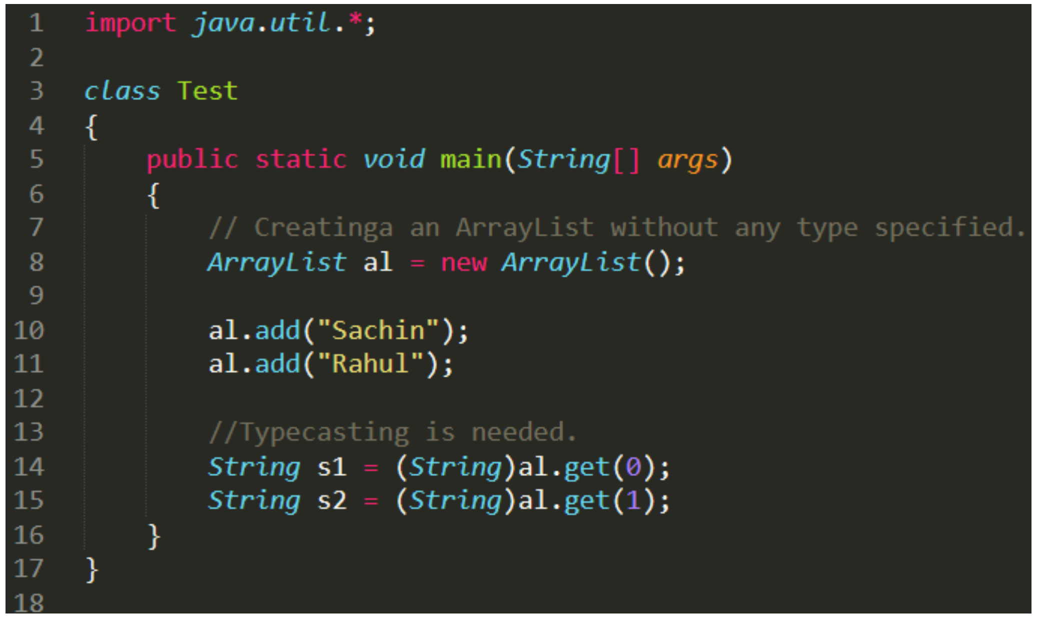
```
public class Array<E> {

    private E[] data;
    private int size;

    public Array(int capacity){
        data = (E[])new Object[capacity];
        size = 0;
    }

    public Array(){
        this(10);
    }

    //Add an element located at the specific index
    public void add(int index, E e){
        for(int i = size - 1; i >= index ; i --)
            data[i + 1] = data[i];
        data[index] = e;
        size ++;
    }

    //Remove an element located at the specific index and return the value.
    public E remove(int index){
        E ret = data[index];
        for(int i = index + 1 ; i < size ; i ++)
            data[i - 1] = data[i];
        size --;
        data[size] = null;
        return ret;
    }
}
```

- **Code Reuse:** We can implement a method/class/interface once and use for any type we want.

Generics in Java 2

Why use generics?

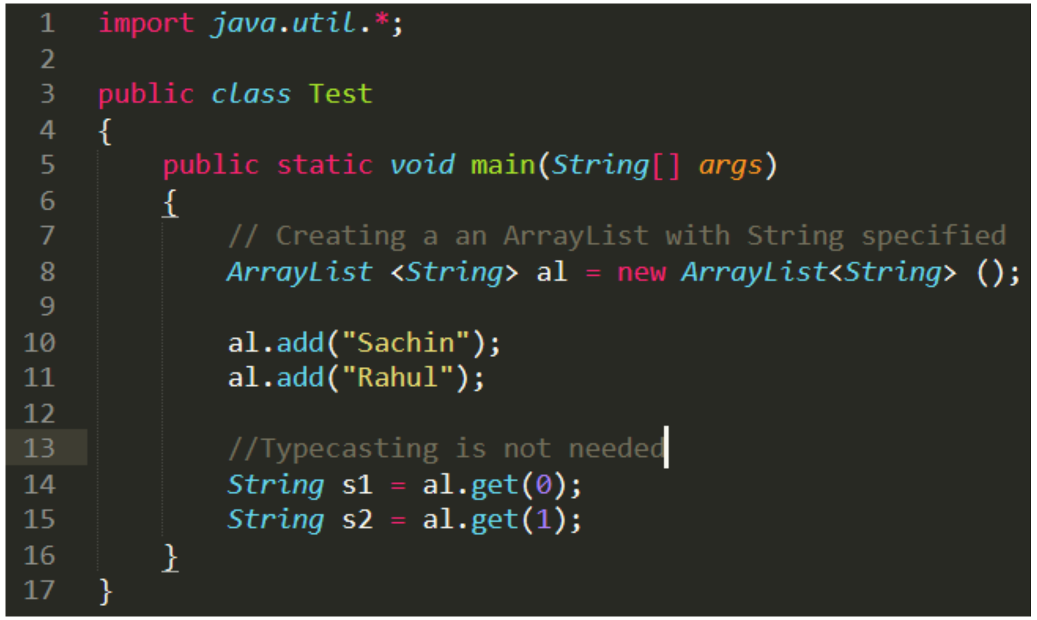
non-generic code



```
import java.util.*;

class Test
{
    public static void main(String[] args)
    {
        // Creatinga an ArrayList without any type specified.
        ArrayList al = new ArrayList();

        al.add("Sachin");
        al.add("Rahul");

        //Typecasting is needed.
        String s1 = (String)al.get(0);
        String s2 = (String)al.get(1);
    }
}
```

VS.

generic code



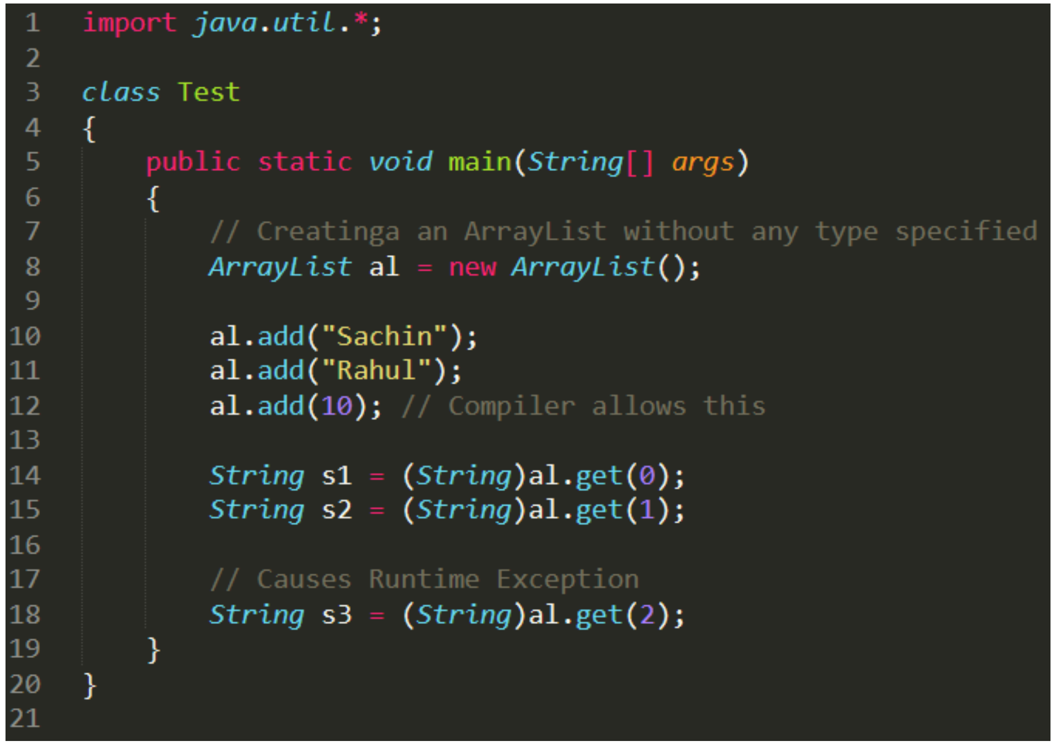
```
import java.util.*;

public class Test
{
    public static void main(String[] args)
    {
        // Creating a an ArrayList with String specified
        ArrayList <String> al = new ArrayList<String> ();

        al.add("Sachin");
        al.add("Rahul");

        //Typecasting is not needed
        String s1 = al.get(0);
        String s2 = al.get(1);
    }
}
```

- **Elimination of casts:** Typecasting at every retrieval operation is a big headache. If we already know that our list only holds a certain type of data then we need not to typecast it every time.

Generics in Java 3

Why use generics?

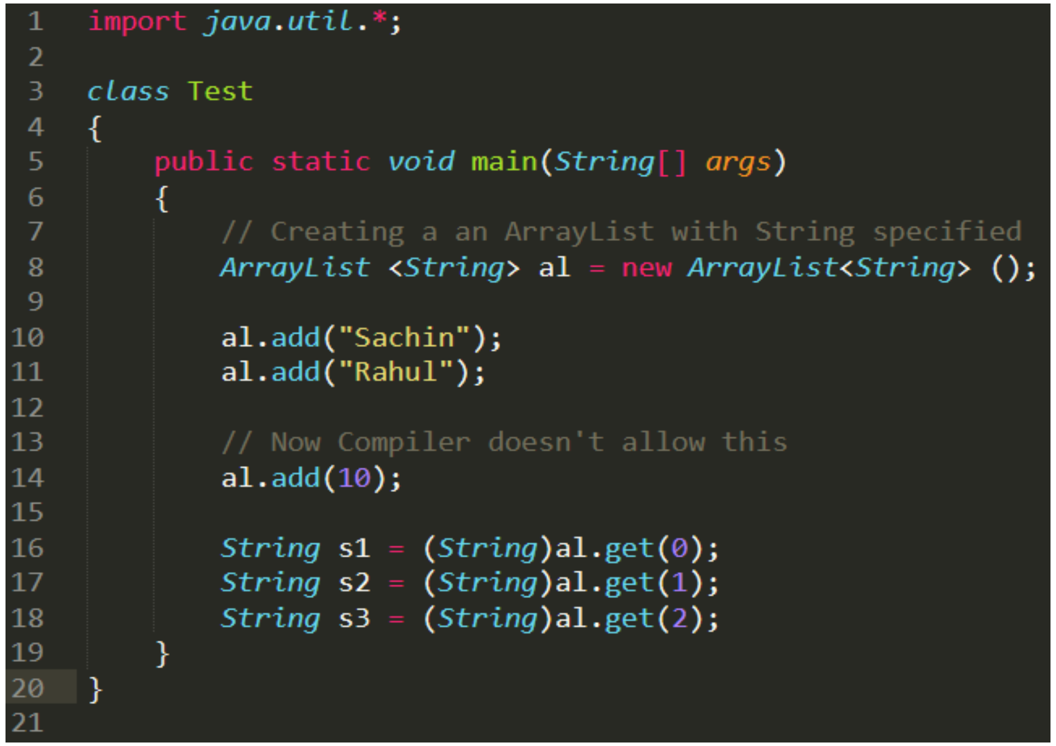
non-generic code



```java
import java.util.*;

class Test
{
    public static void main(String[] args)
    {
        // Creatinga an ArrayList without any type specified
        ArrayList al = new ArrayList();

        al.add("Sachin");
        al.add("Rahul");
        al.add(10); // Compiler allows this

        String s1 = (String)al.get(0);
        String s2 = (String)al.get(1);

        // Causes Runtime Exception
        String s3 = (String)al.get(2);
    }
}

```

VS.

generic code



```java
import java.util.*;

class Test
{
    public static void main(String[] args)
    {
        // Creating a an ArrayList with String specified
        ArrayList <String> al = new ArrayList<String> ();

        al.add("Sachin");
        al.add("Rahul");

        // Now Compiler doesn't allow this
        al.add(10);

        String s1 = (String)al.get(0);
        String s2 = (String)al.get(1);
        String s3 = (String)al.get(2);
    }
}

```

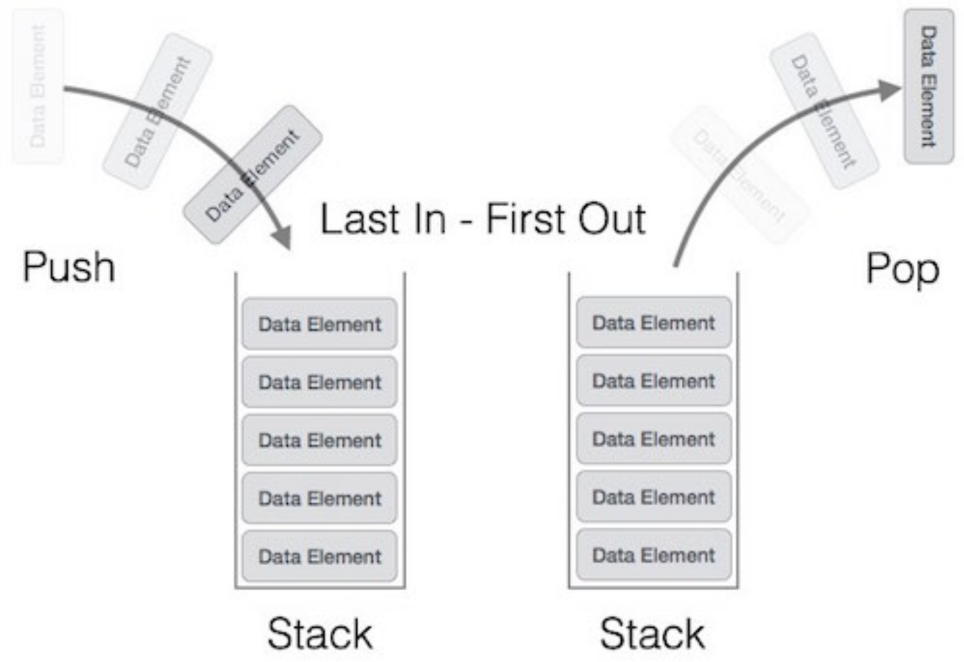
- **Type Safety :** Generics make errors to appear compile time than at run time (Finding bugs in compile-time can save time for debugging java program, because compile-time bugs are much easier to find and fix).

Built-in Data Structures for Java

Stack: a linear data structure which follows Last In First Out (LIFO) order in which the operations are performed.

Methods in Stack class:

- **Object push(Object element)** : Pushes an element on the top of the stack.
- **Object pop()** : Removes and returns the top element of the stack.
- **Object peek()** : Returns the element on the top of the stack, but does not remove it.
- **boolean empty()** : It returns true if nothing is on the top of the stack. Else, returns false.



Java Collection framework provides a Stack class which models and implements Stack data structure.

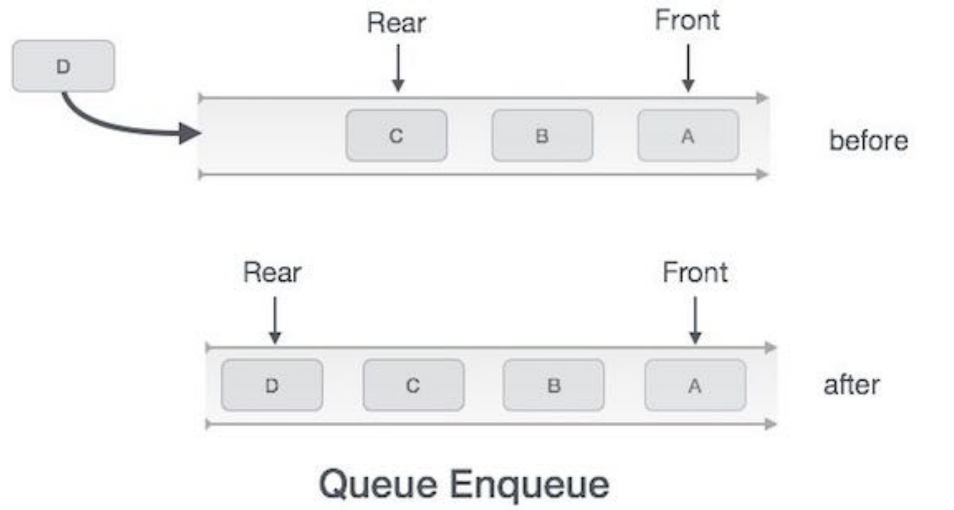
Built-in Data Structures for Java

Queue: a linear data structure which follows First In First Out (FIFO) order in which the operations are performed.

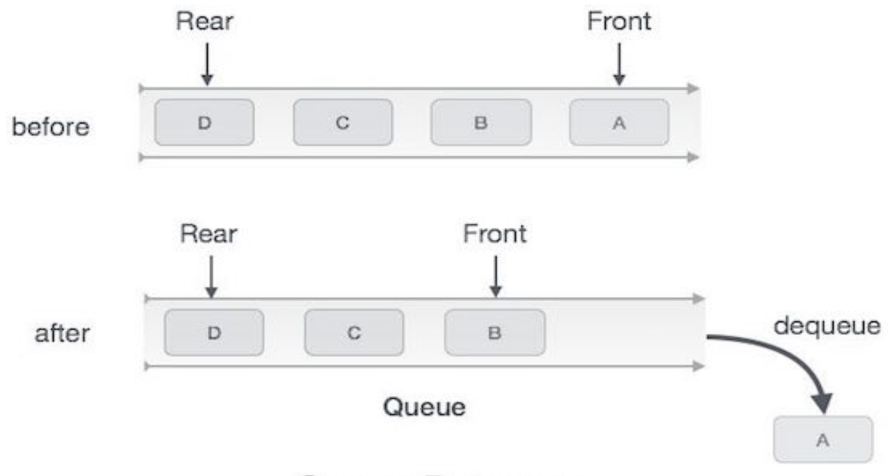
Methods in Queue:

- **Object add(Object element)-** Adds elements at the tail of queue.
- **Object remove()** : Removes and returns the head of the queue.
- **Object peek()** : Returns the head of queue without removing it.
- **Object poll()** : Removes and returns the head of the queue

The Queue interface is available in java.util package and extends the Collection interface



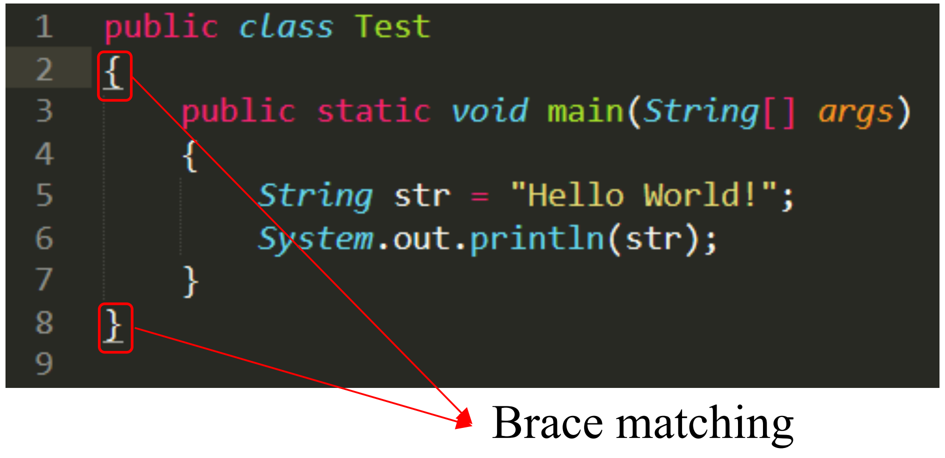
Queue Enqueue



Queue Dequeue

Why Learning Data Structures?

Real World Application: Bracket (brace, parenthesis) matching is an often-used feature almost provided by any IDEs. How it works?

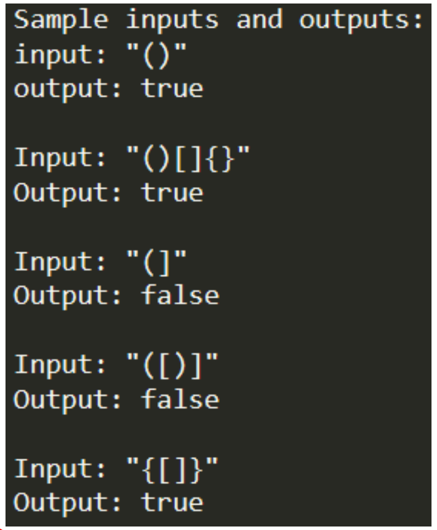


```
public class Test
{
    public static void main(String[] args)
    {
        String str = "Hello World!";
        System.out.println(str);
    }
}
```

Coding Challenge:

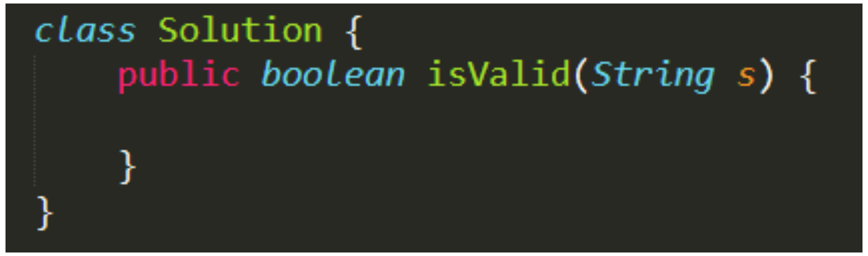
Given a string containing just the characters '(', ')', '{', '}', '[' and ']', determine if the input string is valid.

An input string is valid if:
1.Open brackets must be closed by the same type of brackets.
2.Open brackets must be closed in the correct order.
Note that an empty string is also considered valid.



```
Sample inputs and outputs:
input: "()"
output: true

Input: "()[]{}"
Output: true

Input: "(]"
Output: false

Input: "([)]"
Output: false

Input: "{[]}"
Output: true
```

Please Implement this method:



```
class Solution {
    public boolean isValid(String s) {

    }
}
```

Hint: Using Stack data structure.