Data Structures

02.10.2022

GREEN FOX ACADEMY

Primitive data type vs object

Properties	Primitive data types	Objects
Origin	Pre-defined data types	User-defined data types
When copied	Two different variables is created along with different assignment(only values are same)	Two reference variable is created but both are pointing to the same object on the heap
When changes are made in the copied variable	Change does not reflect in the original ones.	Changes reflected in the original ones.
Default value	Primitive datatypes do not have null as default value	The default value for the reference variable is null
Example	byte, short, int, long, float, double, char, boolean	array, string class, interface etc.

Primitives

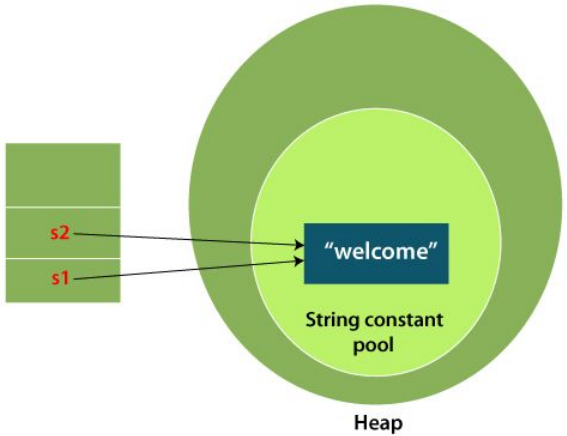
- **byte :** It is 1 byte(8-bits) integer data type. Value range from -128 to 127. Default value zero. example: byte b=10;
- **short :** It is 2 bytes(16-bits) integer data type. Value range from -32768 to 32767. Default value zero. example: short s=11;
- **int :** It is 4 bytes(32-bits) integer data type. Value range from -2147483648 to 2147483647. Default value zero. example: int i=10;
- **long :** It is 8 bytes(64-bits) integer data type. Value range from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807. Default value zero. example: long l=100012;
- **float :** It is 4 bytes(32-bits) float data type. Default value 0.0f. example: float ff=10.3f;
- **double:** It is 8 bytes(64-bits) float data type. Default value 0.0d. example: double db=11.123;
- **char:** It is 2 bytes(16-bits) unsigned unicode character. Range 0 to 65,535.
- **boolean:** Boolean type is used when we want to test a particular condition during the execution of the program. There are only two values that a Boolean type can take: true or false. (1 bit) default is false.
- For example, default Object size is 16 bytes. Object Integer is 16 bytes but int is only 4 bytes.

String

- The **String** class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class.
- It consists of array of characters.
    - **char**[] ch={‘G’, ‘F’, ’A’};
    - String s1=**new** String(ch);
    - String s2="GFA";
- Strings are **constant, their values cannot be changed after they are created**.
- The class String includes methods for examining individual characters of the sequence, for comparing strings, for searching strings, for extracting substrings, and for creating a copy of a string with all characters translated to uppercase or to lowercase.

String

- Each time you create a string literal, the JVM checks the "string constant pool" first. If the string already exists in the pool, a reference to the pooled instance is returned. If the string doesn't exist in the pool, a new string instance is created and placed in the pool. For example:
  - String s1="*GFA*";
  - String s2="*GFA*"; //It doesn't create a new instance
- In the above example, only one object will be created.
- Firstly, JVM will not find any string object with the value
- "GFA" in string **constant pool** that is why it will
- create a new object. After that it will find the string with
- the value "GFA" in the pool, it will not create a new
- object but will return the reference to the same instance.



StringBuilder

- **StringBuilder** in Java represents a mutable sequence of characters. Since the String Class in Java creates an immutable sequence of characters, the StringBuilder class provides an alternate to String Class, as it creates a mutable sequence of characters.
- StringBuilder str = new StringBuilder();
- str.append("GFG");

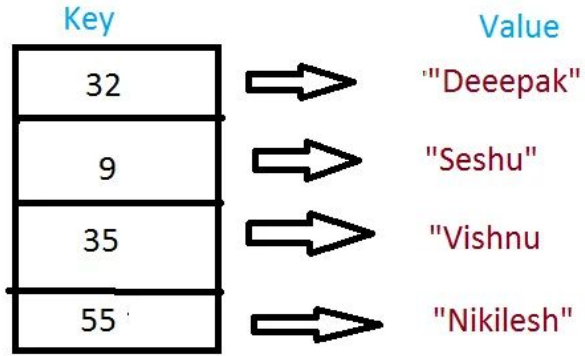
ArrayList

- Ordered
- Indexed
- from 0

Criteria	ArrayList	Array
Resizable	ArrayList is a resizable array.	Array is a fixed length data structure. We cannot change length of array once created in Java
Creating instance with/without specifying size.	You can create instance of ArrayList without specifying size, Java will create Array List with default size .	its mandatory to provide size of Array while creating either directly or indirectly by initializing Array while creating it.
Primitives	We cannot store primitives in ArrayList, it can only store objects.	Array can contain both primitives and objects in Java.
Operations	ArrayList supports many additional operations like indexOf(), remove(), etc.	These functions are not supported by Arrays.
Generics	ArrayList allows you to use Generics to ensure type-safety.	You can not use Generics with Array.
Length	Length of the ArrayList is provided by the size() method	Array object has the length variable which returns the length of the array.
Multi-dimensional	ArrayList doesn't allow you to specify dimension.	You can have a two-dimensional array or a three-dimensional array.
Adding elements	We can insert elements into the arraylist object using the add() method	Array use assignment operator to store elements
Flexibility	ArrayList is more flexible than a plain native array because it's dynamic. It can grow itself when needed	Array is a fixed length data structure.

HashMap

- It stores the data in **(Key, Value)** pairs, and you can access them by an index of another type (e.g. an Integer). One object is used as a key (index) to another object (value). If you try to insert the duplicate key, it will replace the element of the corresponding key.
- Map**<Integer**,**String>** map=new HashMap**<>()**; //HashMap creation
- map.put**(**32,"Deepak"**)**;
- map.put**(**9,"Seshu"**)**; //adding values to HashMap
- map.put**(**35,"Vishnu"**)**;
- map.put**(**55,"Nikilesh"**)**;
- Unordered!!
- Indexed by key!
- KEY MUST BE UNIQUE



Thank you for your attention!