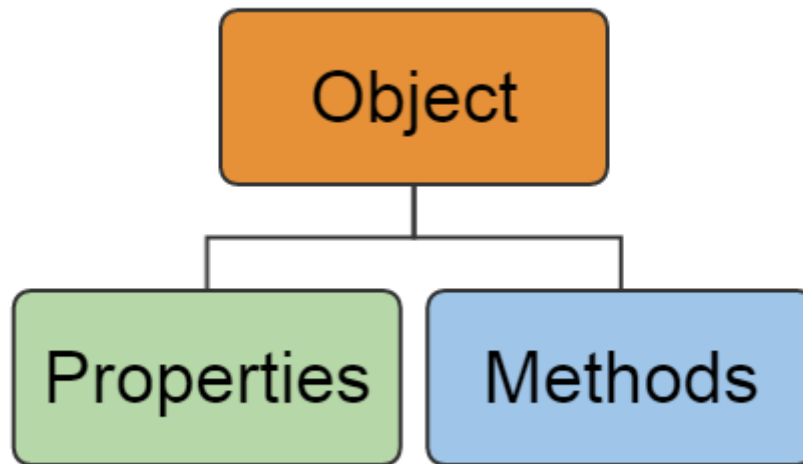




Object Oriented Programming

by Vlad Costel Ungureanu
for "Learn Stuff"



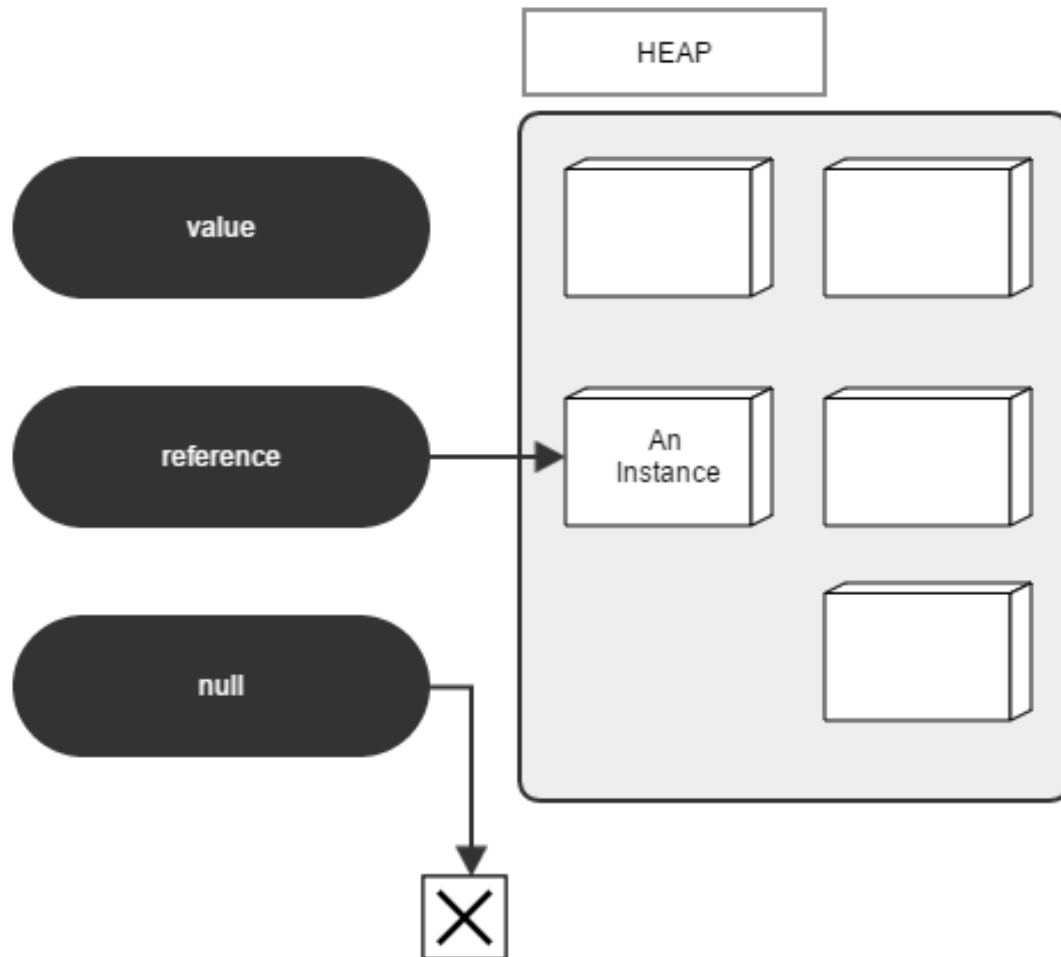
- ✓ Class: Prototype/blueprint which describes objects.
- ✓ Object: Entity described by state and behavior. An object is an instance of a Class.
- ✓ Reference: Entity which provides information for uniquely localizing an object in memory.
- ✓ Program: A multitude of object that interact with one another.
- ✓ Interface: A contract which imposes a specific behavior to a class.
- ✓ Package: A folder name structure necessary for organizing code.

```
[public][abstract][final] class ClassName  
[extends ParentClass]  
[implements Interface1 [, .. ]] {  
    // Variables  
    // Constructors  
    // Methods  
    // Imbricated classes  
}
```

- ✓ Private: only accessible in within the class
- ✓ Protected: only accessible for class and sub classes
- ✓ Public: accessible for everyone
- ✓ Implicit: accessible at package level

```
public class Student {  
    public String name = "";  
    public Student() {};  
    public Student(String studentName) {  
        this.name = studentName;  
    };  
    public void introduceYourself(){  
        System.out.println("Hi, my name is " + this.name);  
    }  
}
```

- ✓ Student studentOne = new Student ("Vlad");
- ✓ Student studentTwo = new Student ();



Object – the Super Class of all Java objects

- ✓ `toString` : returns the representation of the object as a array of characters
- ✓ `equals` : test the equality of two objects of the same type
- ✓ `hashCode` : returns the HASH value corresponding to the object
- ✓ `getClass` : returns the Class the object has been instantiated with
- ✓ `clone` : creates a shallow copy of the object
- ✓ `finalize` : called by GC before deleting the object

[access modifiers] ReturnedType methodName(ParameterType param, ... args)

```
public void introduceYourself(){  
    System.out.println("Hi, my name is " + this.name);  
}
```

```
public void sayHello(){  
    System.out.println("Hola!");  
}
```

```
public class Person{  
    public void sayHello(){  
        System.out.println("Hello");  
    }  
}
```

```
public class SpanishStudent extends Person{  
    @Override  
    public void sayHello(){  
        System.out.println("Hola!");  
    }  
}
```

- ✓ Super classes group common functionality for all subclasses
- ✓ Sub classes use parent defined behavior and define specific behavior

```
class A {  
    public A() {  
        System.out.println("A");  
    }  
}  
  
class B extends A {  
    public B() {  
        System.out.println("B");  
    }  
}  
  
class C extends B {  
    public C() {  
        System.out.println("C");  
    }  
}  
  
C c = new C();
```

```
[public] abstract class AbstractClass{  
    // Abstract Method  
    abstract void anAbstractMethod();  
    // Normal Method  
    void aNormalMethod() { ... }  
}
```

- ✓ Static variables: values stored at class level not at instance level; are the same in all class instances.

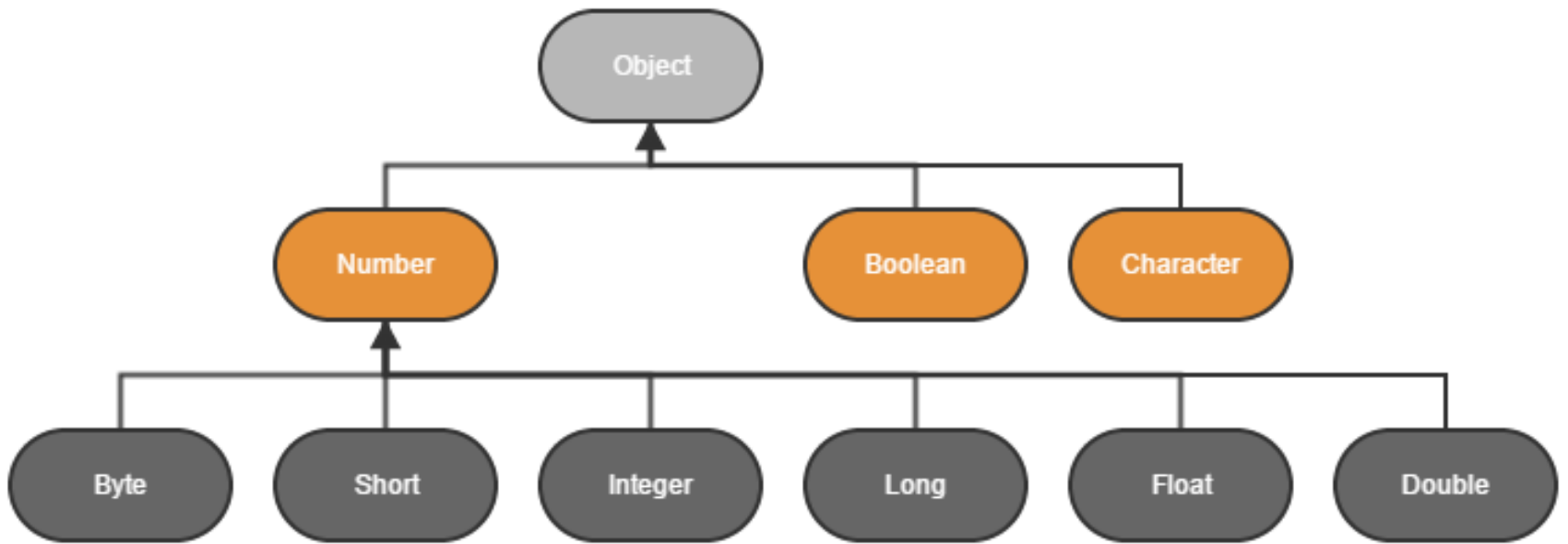
Example: `static final double PI = 3.14;`

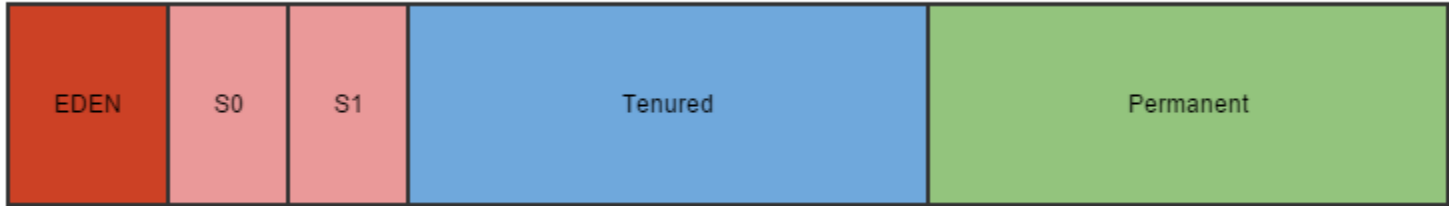
- ✓ Static methods: methods applicable at class level; not applicable at instance level; they can work only with parameters or other static class variables

Example: `double x = Math.sqrt(2);`

- ✓ Static blocks: constructor at class level not instance level; all static blocks are called before any class is instantiated

Example: `static {
 x = 2;
}`





- ✓ Write a class that represents a person. Extend that class for particular types of people: professor, student, janitor, security guard.
- ✓ **Think about** and write common and specific methods for the above classes.
- ✓ Write a class whose instances represent a single playing card from a deck of cards. Playing cards have two distinguishing properties: rank and suit.
- ✓ Write a class whose instances represent a full deck of cards.
- ✓ Write a small program to test your deck and card classes. The program can be as simple as creating a deck of cards and displaying its cards.

Vlad Costel Ungureanu
ungureanu_vlad_costel@yahoo.com

This is a free course from LearnStuff.ro
– not for commercial use –