Globant

Java and OOP Basics

Quality Engineering Studio

Bogotá - Colombia

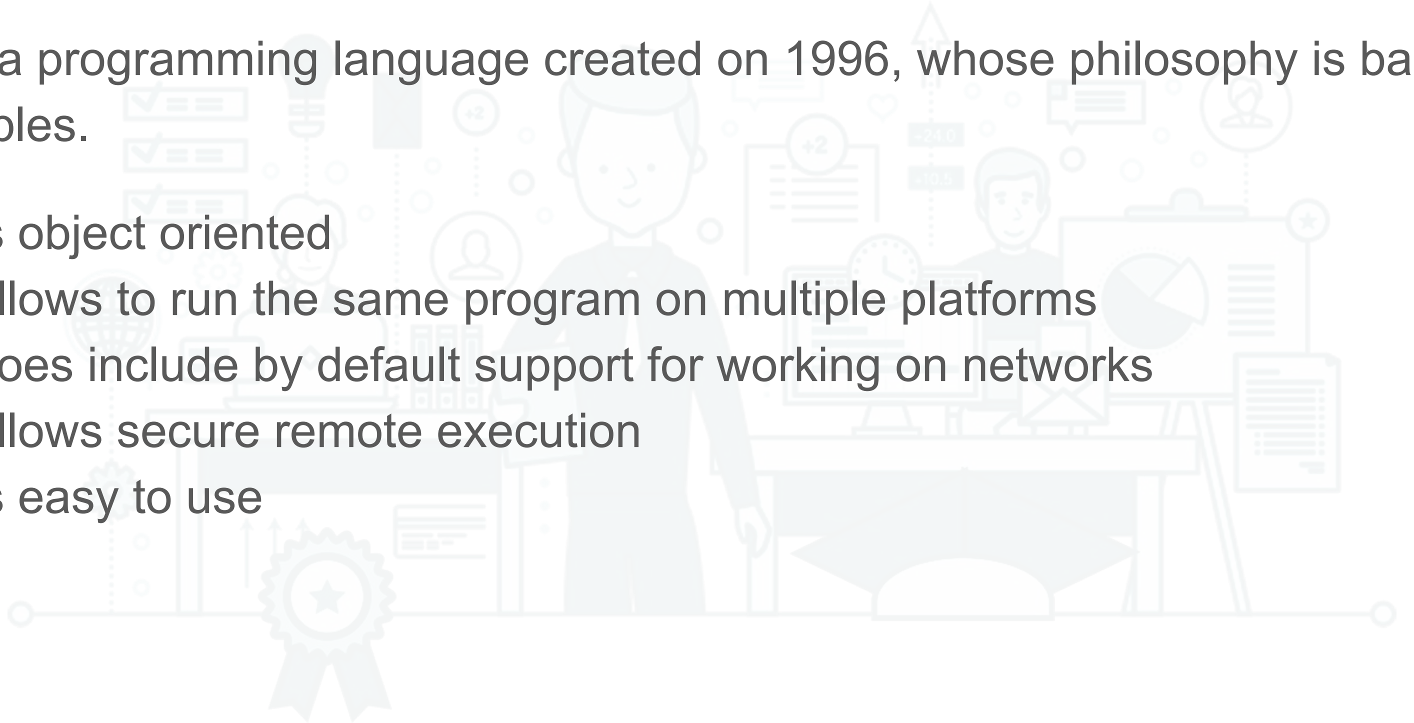
Agenda

- Java Basics
- Java Setup
- Data Types
- OOP
- Classes
- Objects
- Attributes
- Access Modifiers
- Methods
- Main Class
- Read And Print

Java Basics

Java is a programming language created on 1996, whose philosophy is based on 5 principles.

1. It is object oriented
2. It allows to run the same program on multiple platforms
3. It does include by default support for working on networks
4. It allows secure remote execution
5. It is easy to use

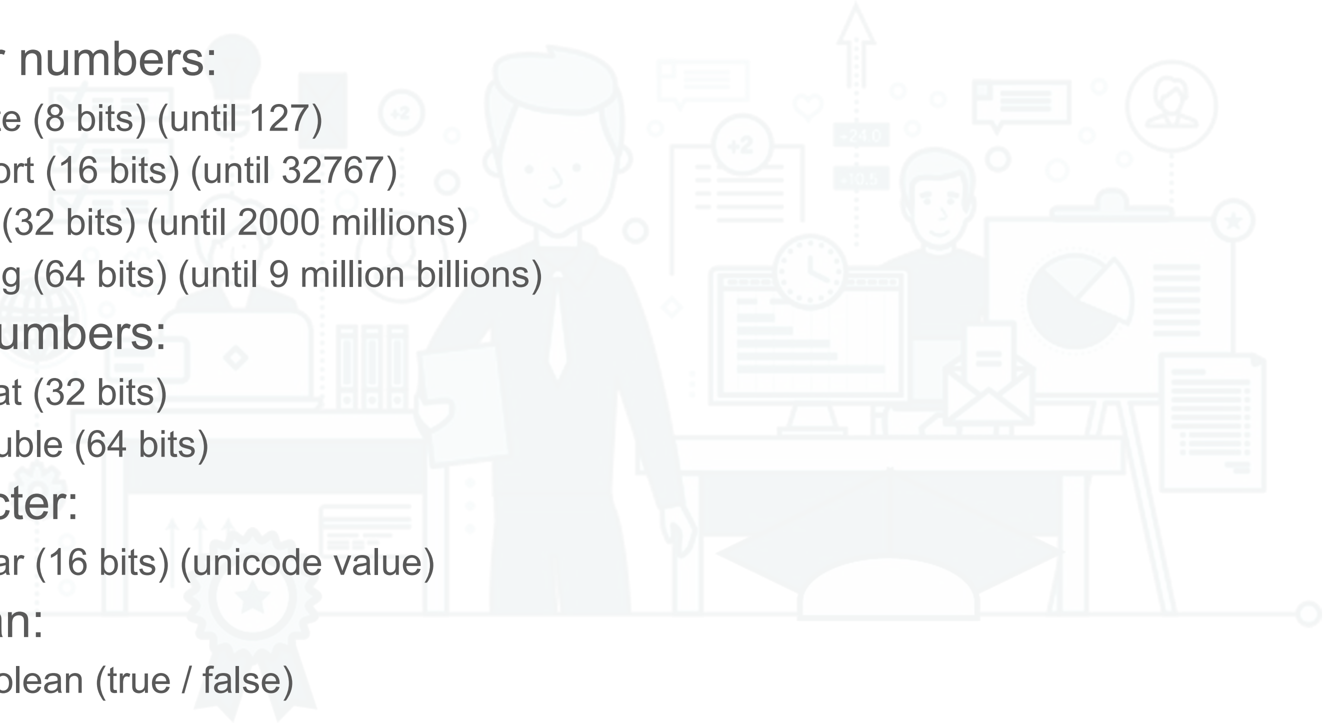


Java Setup

1. Install JDK 8 and JRE 8
2. Add JDK/bin on system PATH
3. Add JAVA_HOME environment variable with JDK route.
4. Add JAVA_HOME to PATH environment variable
5. On CMD run `java -version` and `javac -version` to verify installation
6. Install the IDE

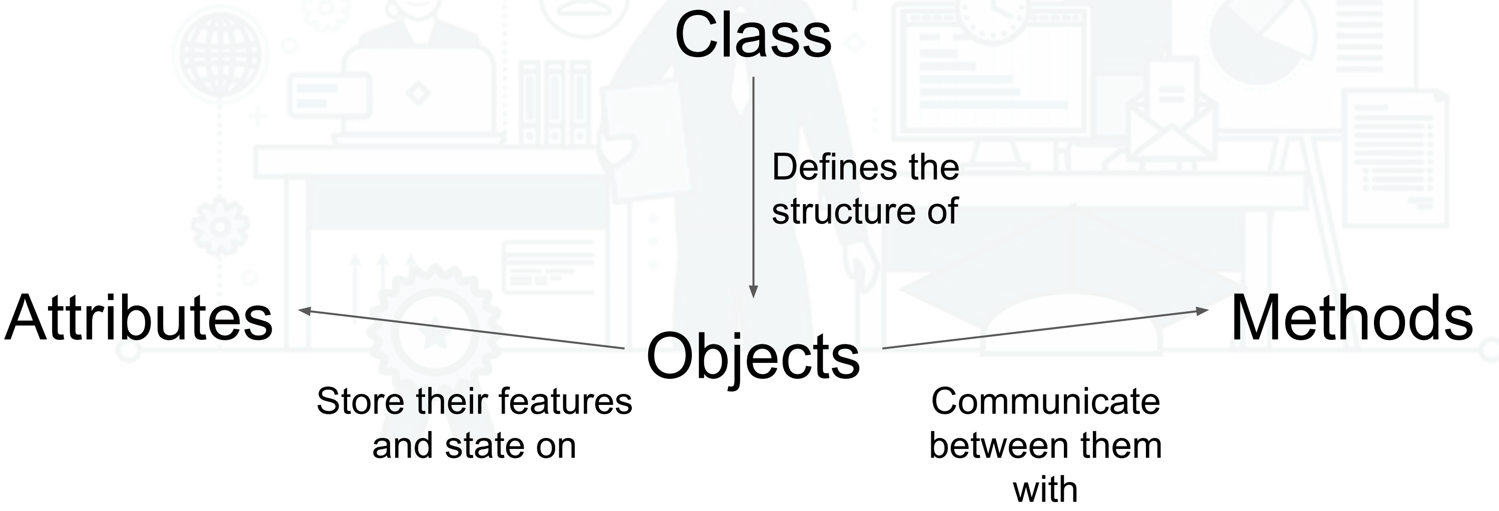
Data Types

- Integer numbers:
  - byte (8 bits) (until 127)
  - short (16 bits) (until 32767)
  - int (32 bits) (until 2000 millions)
  - long (64 bits) (until 9 million billions)
- Real numbers:
  - float (32 bits)
  - double (64 bits)
- Character:
  - char (16 bits) (unicode value)
- Boolean:
  - boolean (true / false)



OOP

Programming paradigm in which objects control data inputs to generate correct data outputs. Programming is done close to an imperative way, but keeping modularity and maintainability.



Class

Defines the structure of

Attributes

Objects

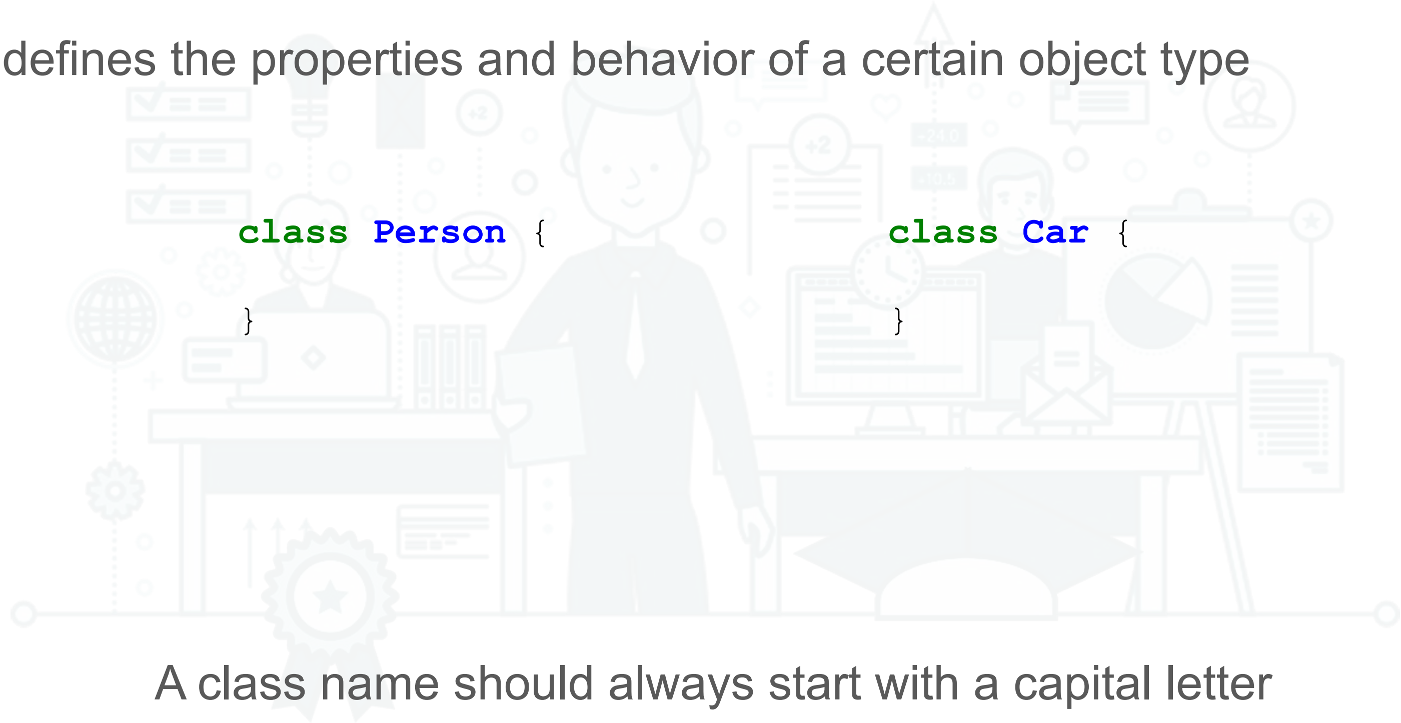
Methods

Store their features and state on

Communicate between them with

Classes

A class defines the properties and behavior of a certain object type

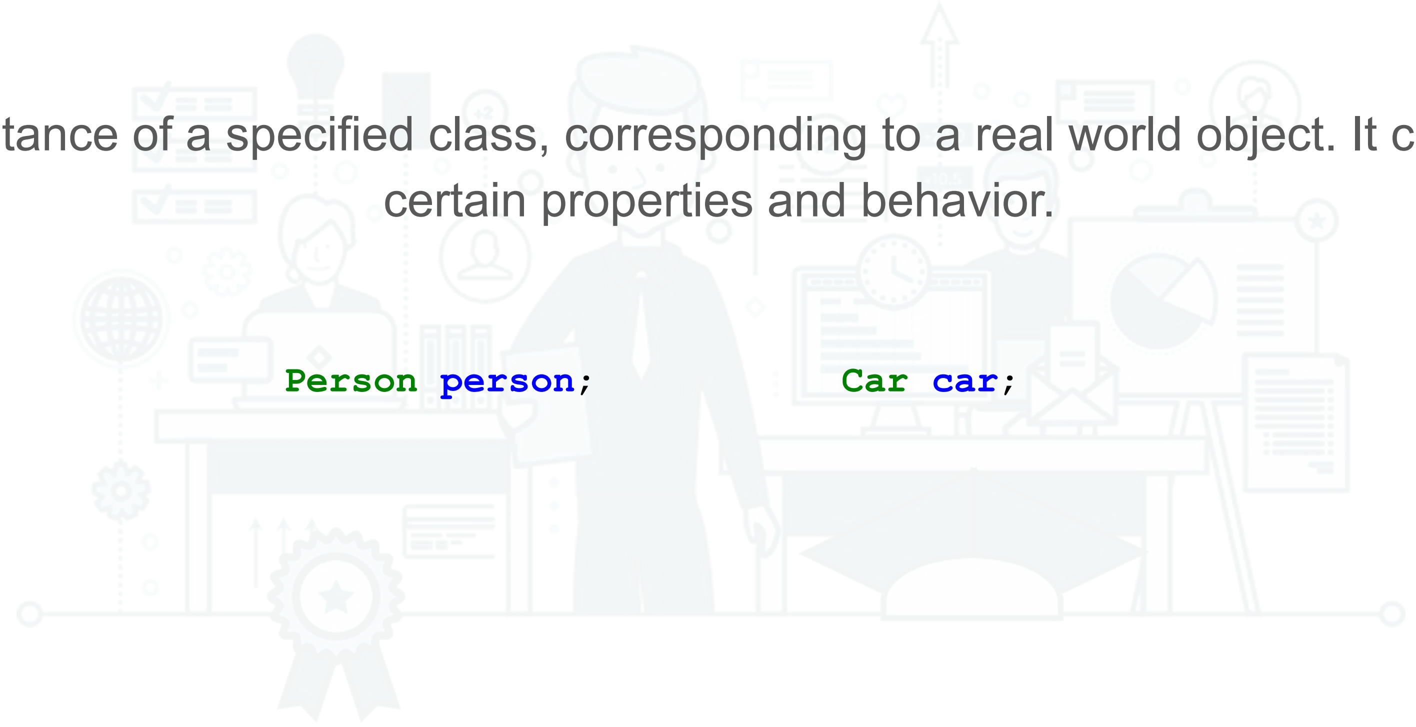


```
class Person {

}
```

```
class Car {

}
```

A class name should always start with a capital letter

Objects



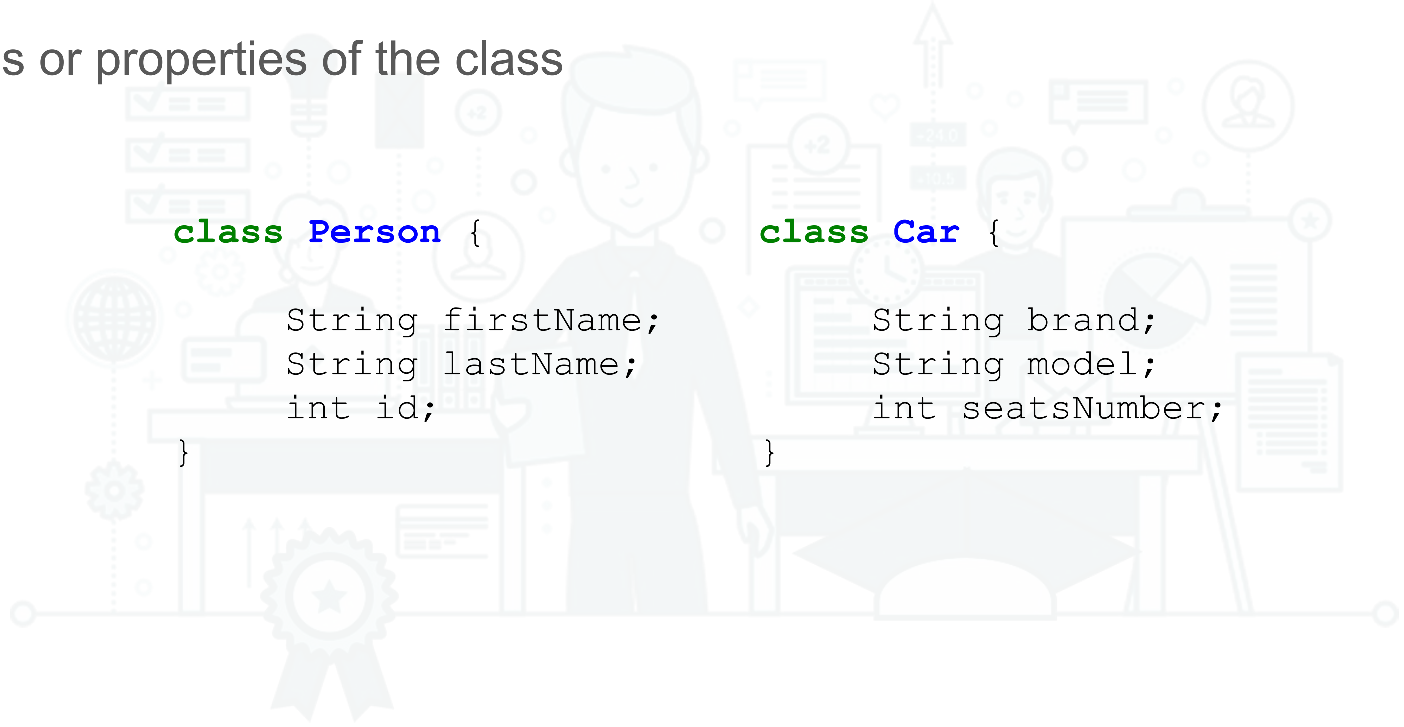
An instance of a specified class, corresponding to a real world object. It contains certain properties and behavior.

```
Person person;
```

```
Car car;
```

Attributes

Features or properties of the class



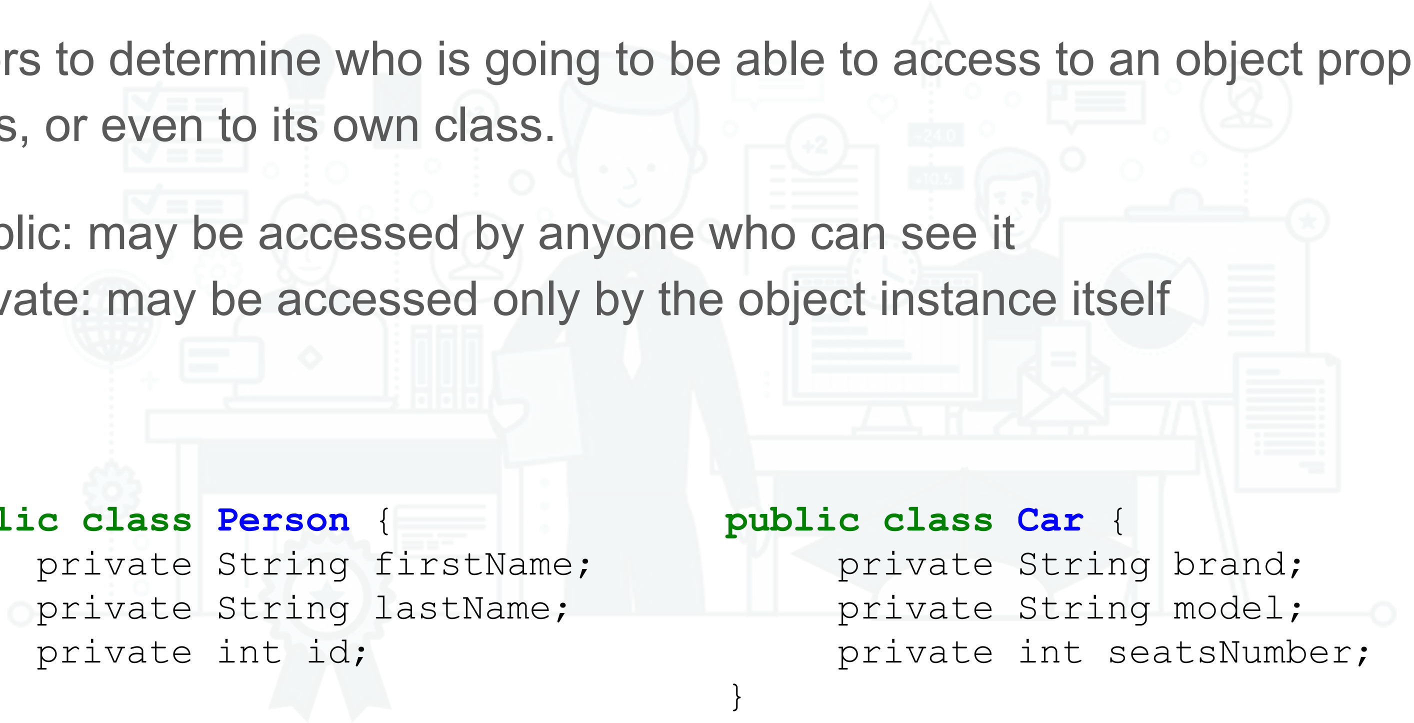
```
class Person {

    String firstName;
    String lastName;
    int id;
}
```

```
class Car {

    String brand;
    String model;
    int seatsNumber;
}
```

Access Modifiers

Indicators to determine who is going to be able to access to an object properties, methods, or even to its own class.

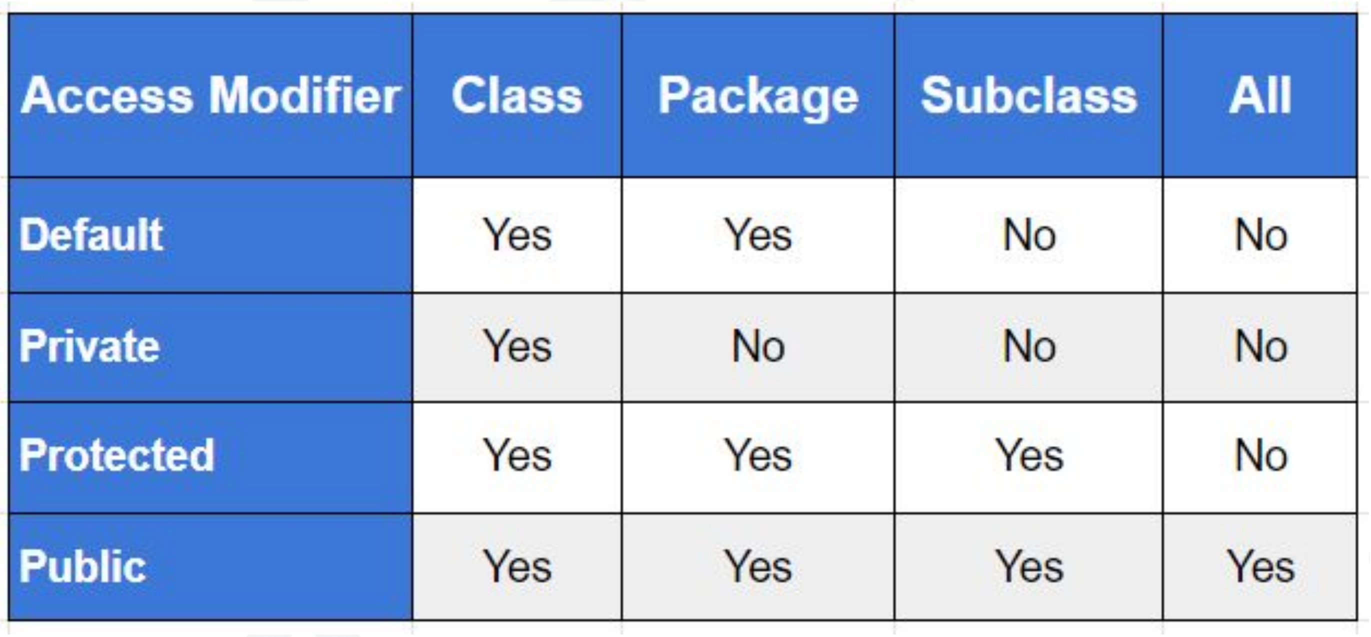
- Public: may be accessed by anyone who can see it
- Private: may be accessed only by the object instance itself



```java
public class Person {
     private String firstName;
     private String lastName;
     private int id;
}
```

```java
public class Car {
     private String brand;
     private String model;
     private int seatsNumber;
}
```

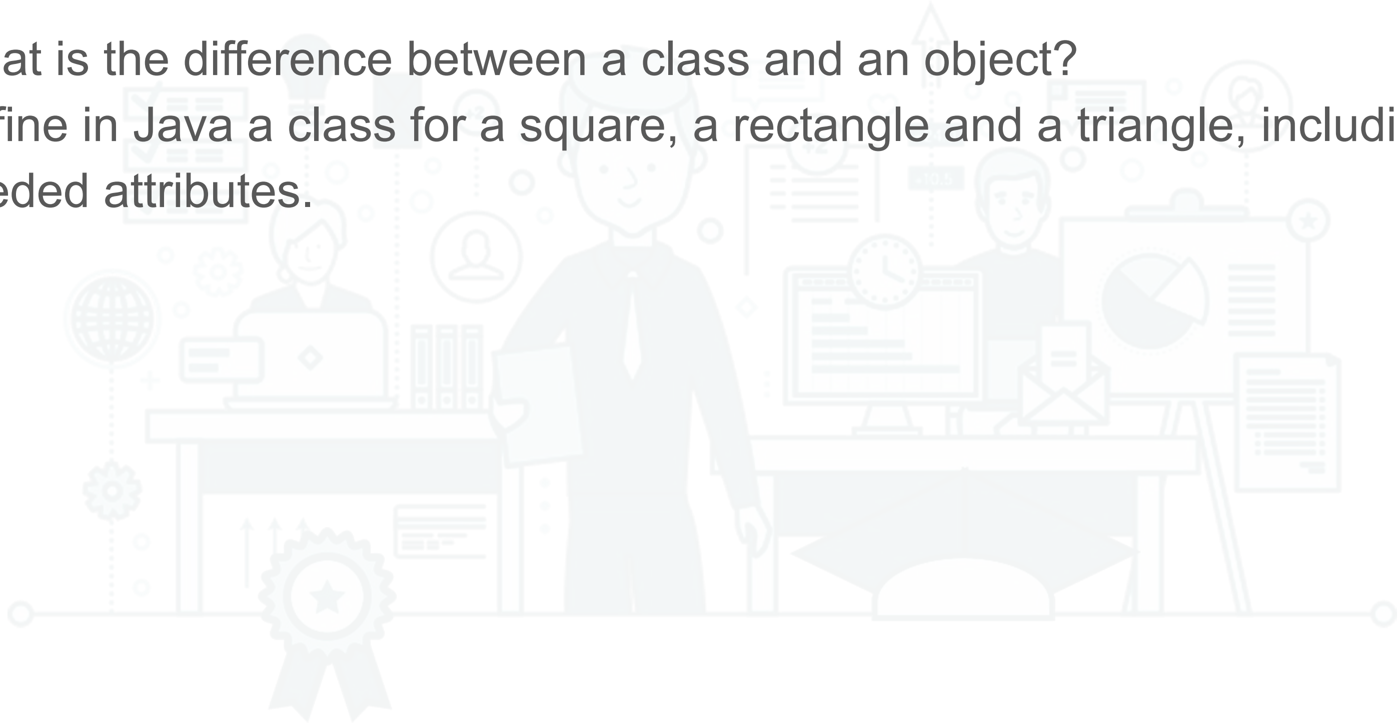
Access Modifier



Access Modifier	Class	Package	Subclass	All
Default	Yes	Yes	No	No
Private	Yes	No	No	No
Protected	Yes	Yes	Yes	No
Public	Yes	Yes	Yes	Yes

Exercise

- What is the difference between a class and an object?
- Define in Java a class for a square, a rectangle and a triangle, including the needed attributes.

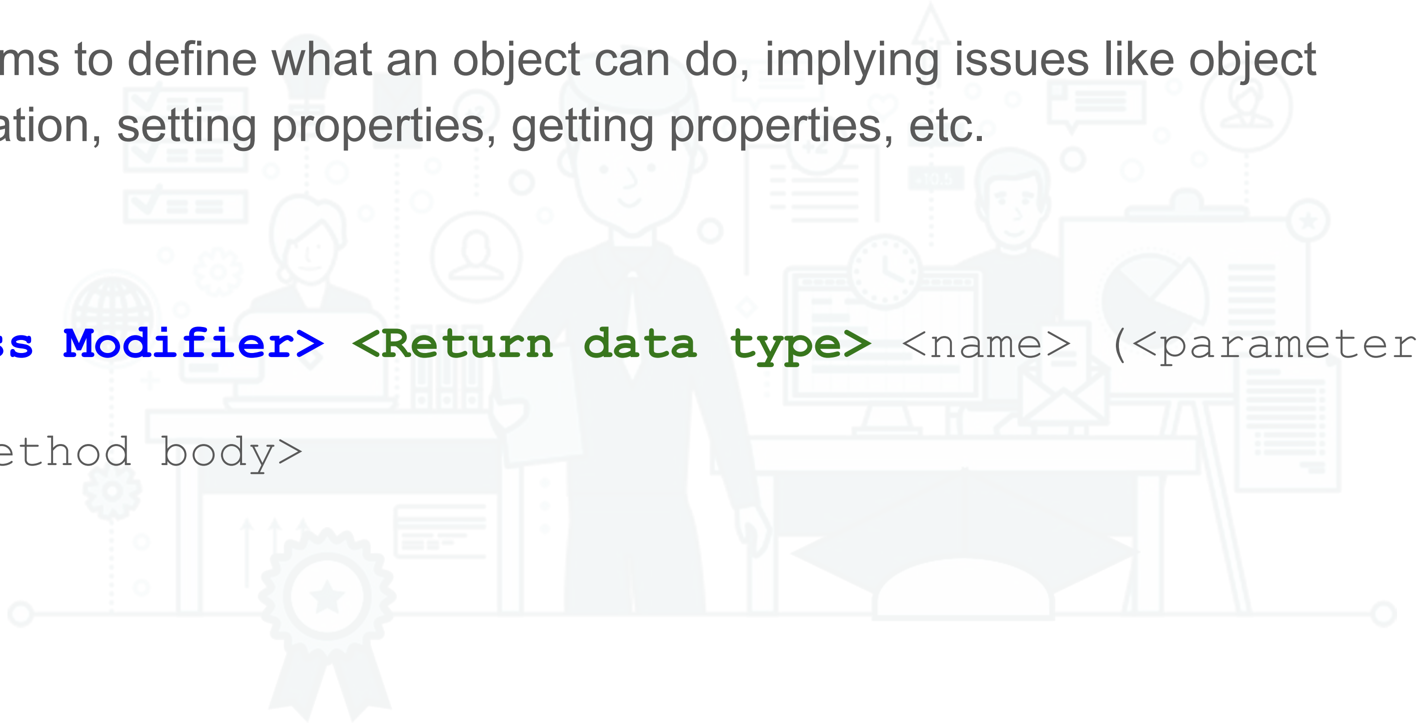


Methods

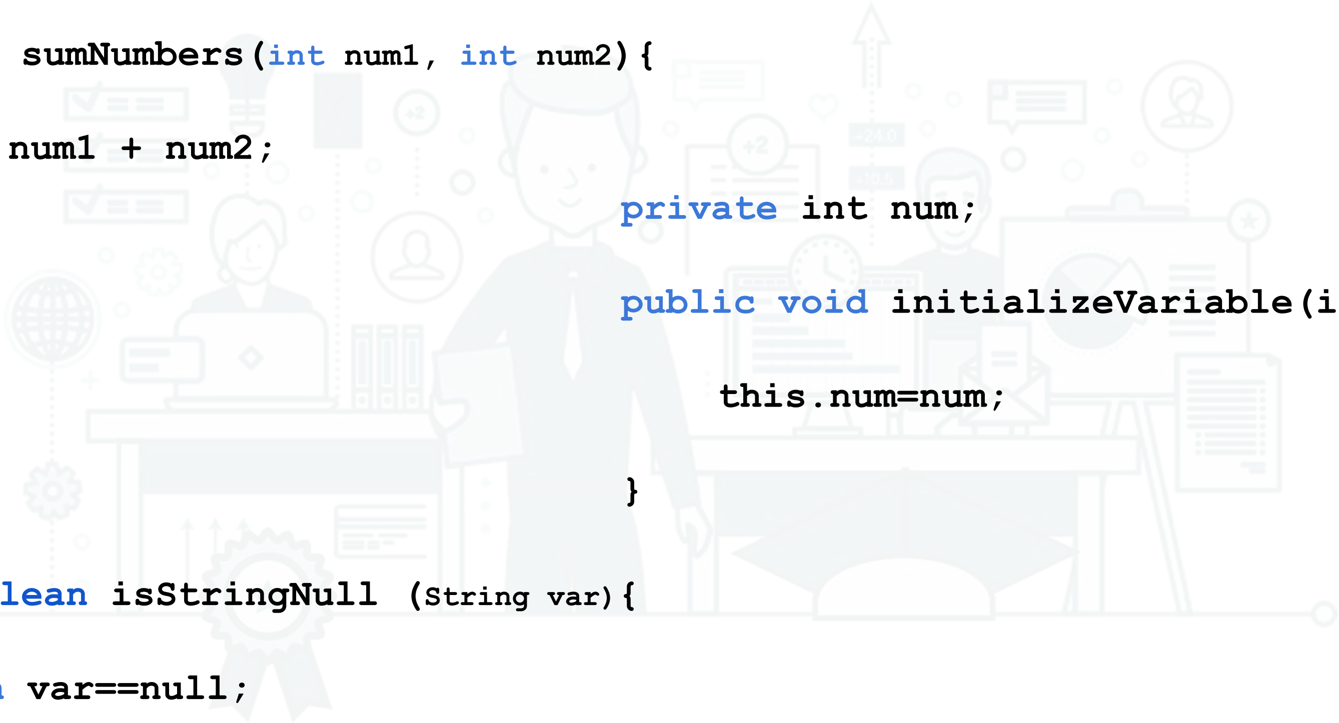
Algorithms to define what an object can do, implying issues like object instantiation, setting properties, getting properties, etc.

Form:

```
<Access Modifier> <Return data type> <name> (<parameters>){

    <Method body>

}
```



Examples



```
public int sumNumbers(int num1, int num2){

    return num1 + num2;

}
```

```
private int num;

public void initializeVariable(int num){

    this.num=num;

}
```

```
public boolean isStringNull (String var){

    return var==null;

}
```

Main Class

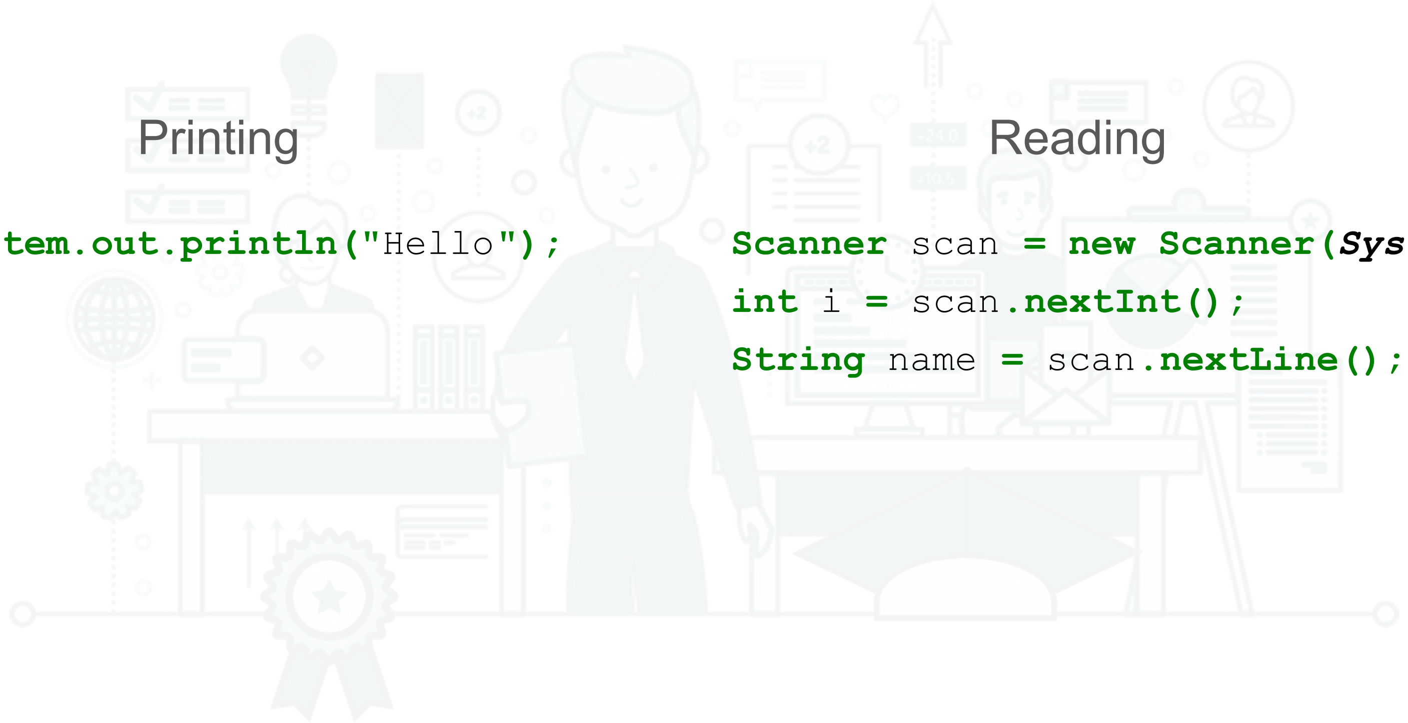
Class where the defined classes and objects are going to be tested or executed

```
public class Test {
   public static void main(String[] args) {
      //main code goes here
   }
}
```

```
public class Runner {
   public static void main(String[] args) {
      //main code goes here
   }
}
```

```
public class Main {
   public static void main(String[] args) {
          //main code goes here
   }
}
```

Reading and Printing on Java



Printing

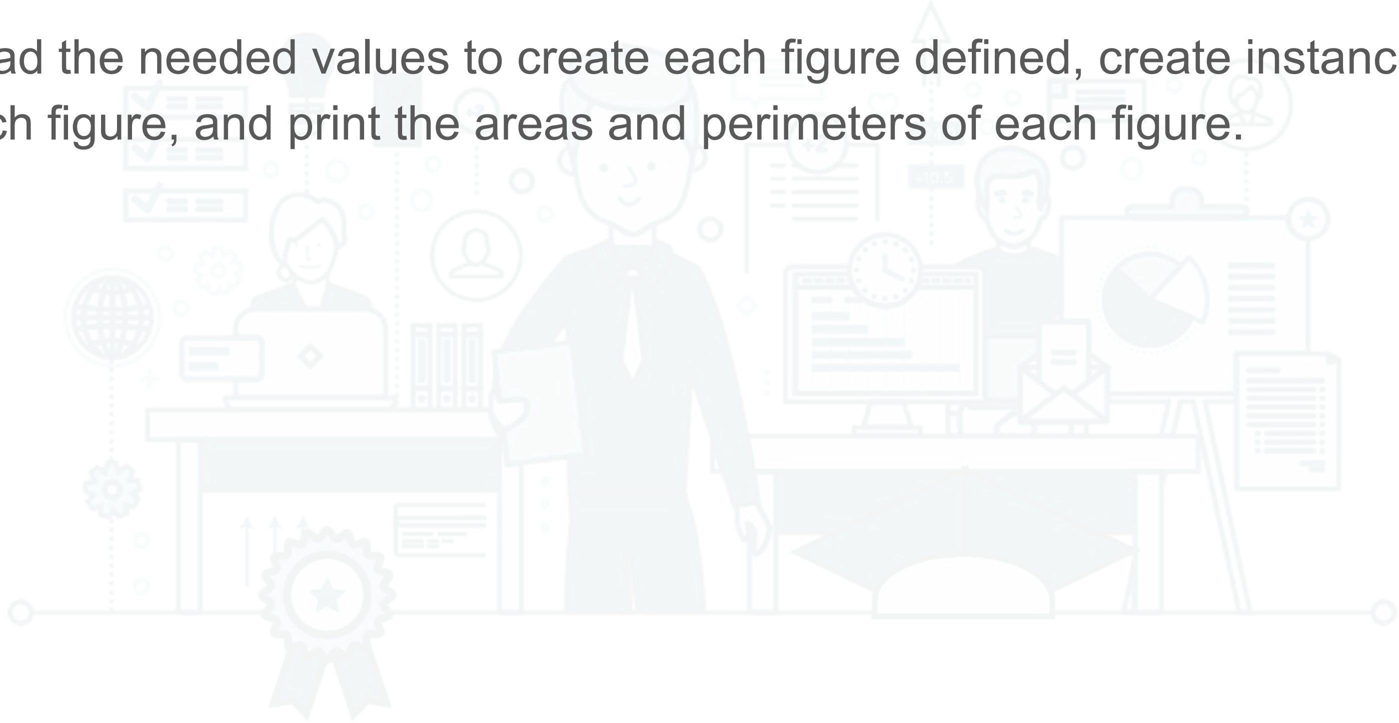
```java
System.out.println("Hello");
```

Reading

```java
Scanner scan = new Scanner(System.in);
int i = scan.nextInt();
String name = scan.nextLine();
```

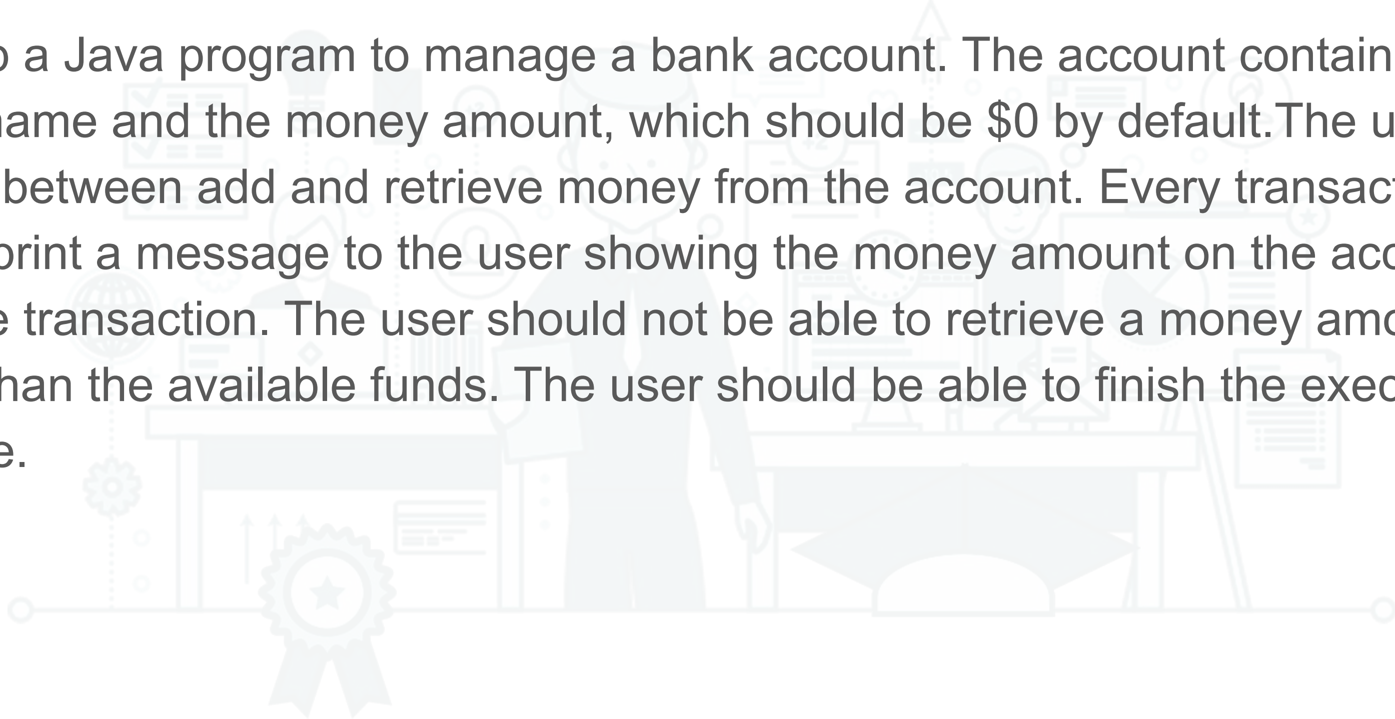
Exercise

- Read the needed values to create each figure defined, create instances of each figure, and print the areas and perimeters of each figure.



Homework

Develop a Java program to manage a bank account. The account contains the owner name and the money amount, which should be $0 by default.The user may choose between add and retrieve money from the account. Every transaction should print a message to the user showing the money amount on the account after the transaction. The user should not be able to retrieve a money amount higher than the available funds. The user should be able to finish the execution any time.



Q&A