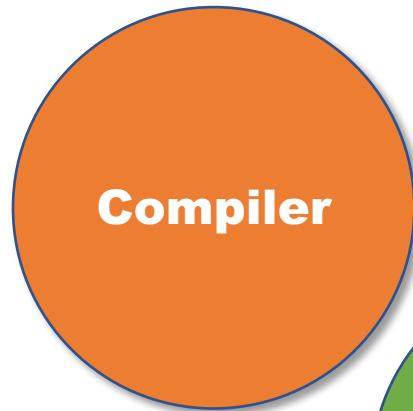
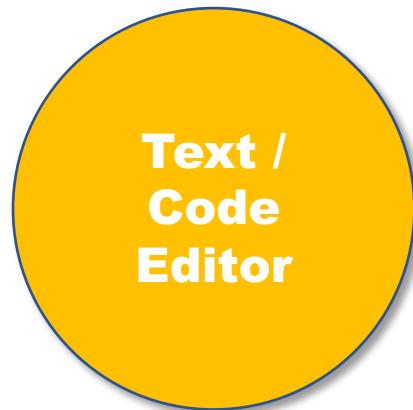


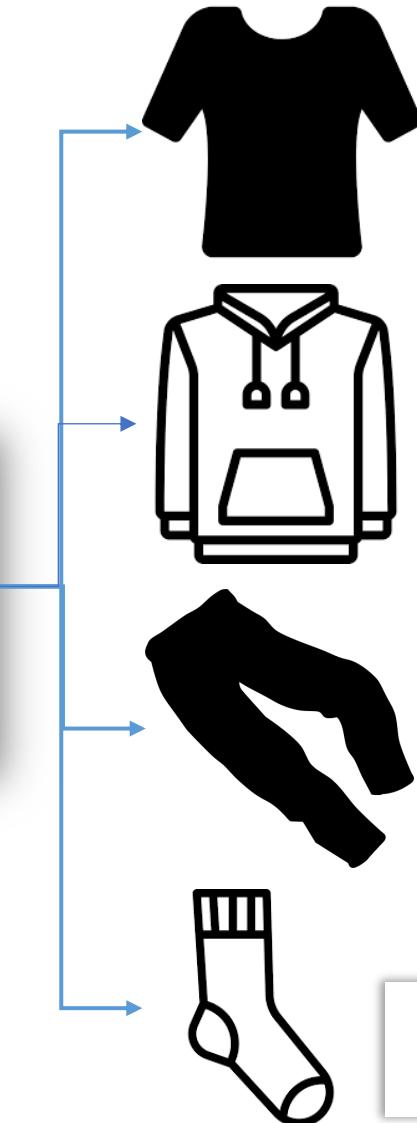
Java Tutorial



IDE – Integrated Development Environment



Variable



Int

Double
Float

Boolean

char

char letter = 'a';



```
int x = 10;  
int y = 67;  
int z = x + y;
```

```
float pi = 3.1415926;  
double long_pi =  
1.61803398875.....7878
```

```
Boolean condition = ( x > y )
```

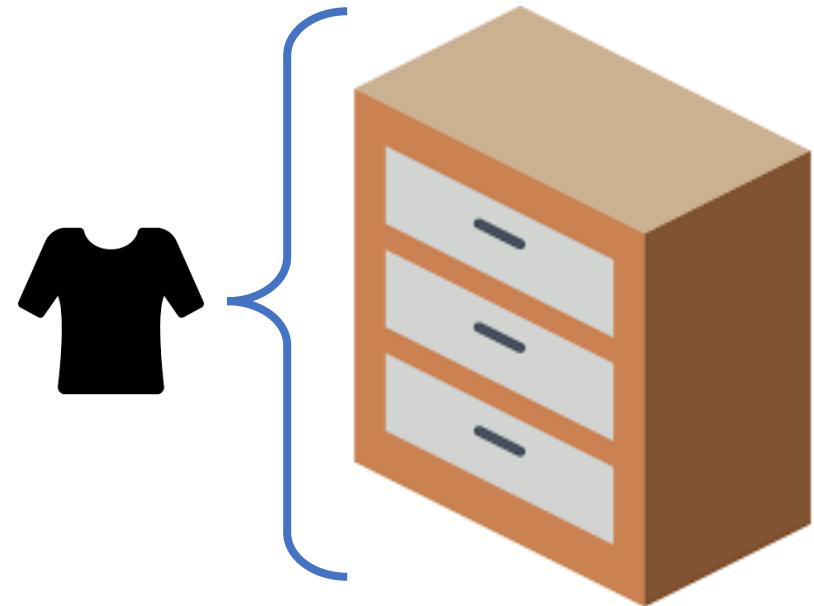
String

```
String word = "Andrzej" =  
[ 'A', 'n', 'd', 'r', 'z', 'e', 'j' ];
```

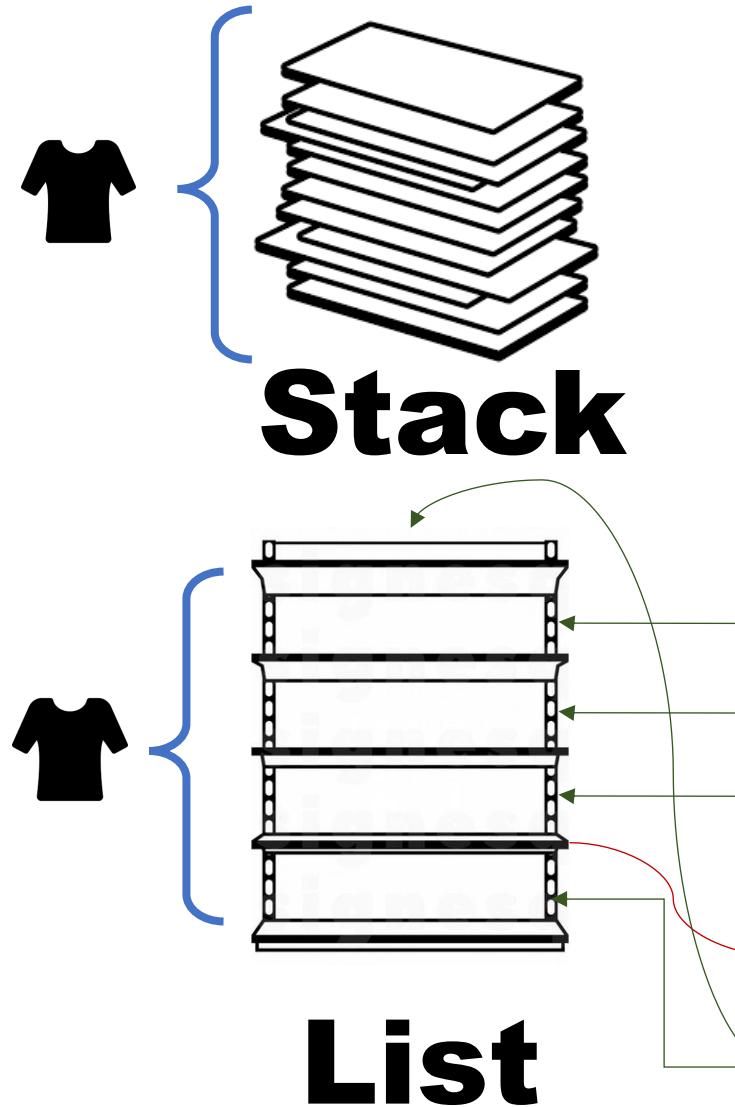
Primitive Types

Type	Bytes	Range
byte	1	[-128, 127]
short	2	[-32K, 32K]
int	4	[-2B, 2B]
long	8	
float	4	
double	8	
char	2	A, B, C, ...
boolean	1	true / false

Arrays, Lists, Maps



Array



List

Take whichever shelf away
Add shelf wherever



Map

The screenshot shows a Java code editor interface with several annotations:

- File structure**: An orange callout points to the left sidebar's tree view of project files.
- Import / libs/ drivers /external files**: A green callout points to the top right corner of the code editor.
- Our own methods**: A green callout points to the `public static void main(String[] args) throws Exception { }` block.
- Variables declaration**: A purple callout points to the variable declarations: `int x = 10;`, `int y = 100;`, and `int z = x + y; // 10 + 100 = 110`.
- MAIN method (function inside the class)**: A yellow callout points to the `main` method definition.
- class**: A light blue callout points to the word "class" in the `public class App { }` declaration.
- terminal**: A light blue callout points to the terminal window at the bottom.

Code Editor Content (App.java):

```
src > App.java > ...
1 import java.util.*;
2
3 public class App {
4     Run | Debug
5     public static void main(String[] args) throws Exception {
6         int x = 10;
7         int y = 100;
8         int z = x + y; // 10 + 100 = 110
9
10        String name = "Andrzej";
11        char letter = 'A';
12        char secondLetter = name.charAt(1);
13
14        System.out.println("Hello, World!"); // output: "Hello World"
15        System.out.println("X = " + x + " | Y = " + y + " | Z = " + z); // output: X = 10 | Y = 100 | Z = 110
16        System.out.println(letter + " | " + secondLetter); // output: A
17        System.out.println("My name is " + name); // output: "My name is Andrzej"
18
19        // INPUTS
20        Scanner sc = new Scanner(System.in); // System.in is a standard input stream.
21
22        System.out.println("What is your name? ");
23        String str = sc.nextLine();
24        System.out.println("What is your age? ");
25    }
}
```

Terminal Output:

```
1: Java Process Console
andrzejswietek@Andrzejs-MacBook-Pro-5 ~/Desktop/JavaTutorialForTigertrons/tuutorial > ls
README.md lib src
andrzejswietek@Anduzej-MacBook-Pro-5 ~/Desktop/JavaTutorialForTigertrons/tuutorial >
```



- > OUTLINE
 - > NPM SCRIPTS
 - > JAVA PROJECTS
- ① 0 △ 0 ① 1 ⌂ Live Share

OOPs

Concepts

Encapsulation

Class

Abstraction

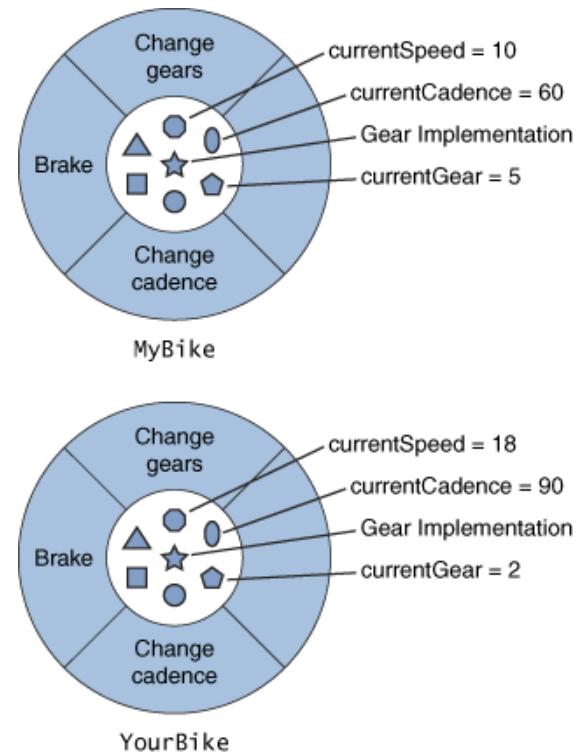
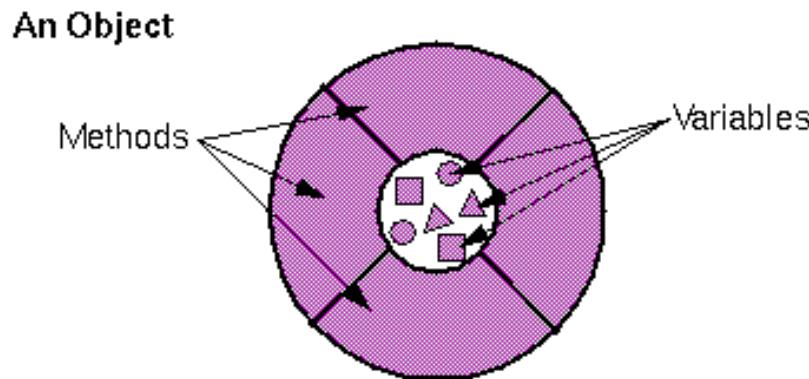
Polymorphism

Object

Inheritance

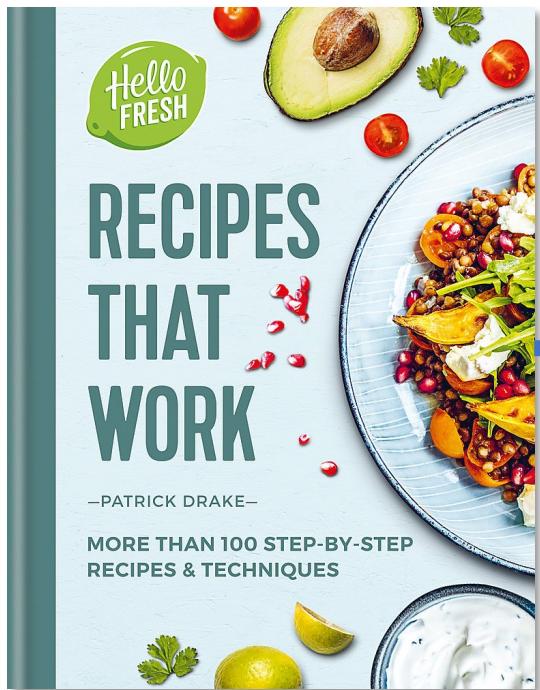
What the object is ???

- ❖ Something we can describe
(therefore it has its own characteristical properties)
 - ❖ We can do something with it.
(methods)



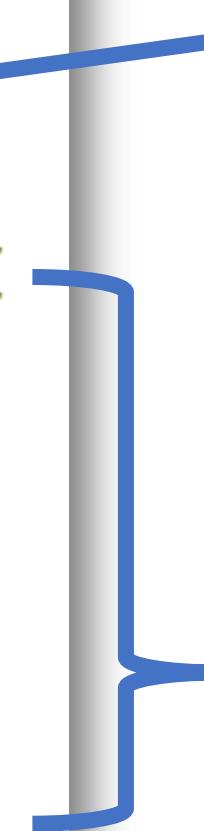
What the class is ???

Collection of data member and member functions
(member stuff declared inside class)



Encapsulation

```
public class DocumentItem {  
  
    private String name;  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```



Inheritance

```
public class Item {  
  
    String name;  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public String getName() {  
        return name;  
    }  
}
```

```
public class DocumentItem extends Item {  
  
    String type;  
  
    public void setType(String type) {  
        this.type = type;  
    }  
  
    public String getType() {  
        return type;  
    }  
}
```

```
public class TextDocumentItem extends DocumentItem {  
}
```

Inheritance

