

Object Oriented Programming Concepts

Supriya Vadiraj

University of Applied Sciences Bonn-Rhein-Sieg

September 3, 2018



**Hochschule
Bonn-Rhein-Sieg**
University of Applied Sciences

Introduction

- Writing object-oriented programs involves creating classes.
- Creating objects from those classes.
- Creating applications, which are stand-alone executable programs that use those objects.



Features in OOPS

- 1 Encapsulation
- 2 Inheritance
- 3 Abstraction
- 4 Polymorphism



Benifits of OOPS

- Code reuse - programmer efficiency
- Encapsulation -code quality, ease of maintenance
- Inheritance - efficiency, extensibility.
- Polymorphism - Robustness of code



Object Oriented Principles -Inheritance

Inheritance can be defined as the process where one class acquires the properties (methods and fields) of another. With the help of inheritance information are managed in a hierarchical manner.

Example:

```
public class Parent {  
  
    public Parent()  
    {  
  
    }  
  
}
```

```
public class Child  
    extends Parent {  
  
    public Child() {  
        super();  
    }  
  
}
```

What does that mean? Among other:

- Child inherits all properties and skills.
- Redundancies can be avoided.



Object Oriented Principles -Polymorphism

- Polymorphism in Java is a concept by which we can perform a single action in different ways.
- There are two types of polymorphism in Java: compile-time polymorphism and runtime polymorphism.
- We can perform runtime polymorphism polymorphism in java by and method overriding and compile-time polymorphism by method overloading.



```
class Shape{
void draw(){System.
    out.println("
        drawing...");}
}
class Rectangle
    extends Shape{
void draw(){System.
    out.println("
        drawing
        rectangle...");}
}
class Circle extends
    Shape{
void draw(){System.
    out.println("
        drawing circle
        ...");}
}
class Triangle
    extends Shape{
```

```
void draw(){System.
    out.println("
        drawing triangle
        ...");}
}
class
    TestPolymorphism2
{
public static void
    main(String args
        []){
    Shape s;
    s=new Rectangle();
    s.draw();
    s=new Circle();
    s.draw();
    s=new Triangle();
    s.draw();
}
}
```



Object Oriented Principles - Abstraction

Abstract classes represent classes that cannot be instantiated. However other classes can inherit from abstract classes and then be instantiated with the properties of the abstract class. It is used as a process of hiding the implementation from the user.

Facts about abstract classes:

- Contain the word *abstract* in their declaration.
- May or may not contain *abstract methods*(=methods whose body still needs to be defined by the heirs).
- If a class has at least one abstract method it needs to be declared abstract.




```
abstract class Bike{
    abstract void run();
}
class Honda4 extends Bike{
void run(){System.out.println("running safely");}
public static void main(String args[]){
    Bike obj = new Honda4();
    obj.run();
}
}
```



Object Oriented Principles - Encapsulation

- Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as a single unit. In encapsulation, the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class. Therefore, it is also known as **data hiding**.
- To achieve encapsulation in Java:
- Declare the variables of a class as private.
- Provide public setter and getter methods to modify and view the variables values.



Object Oriented Principles - Encapsulation

```
/* File name : EncapTest.java */  
public class EncapTest {  
    private String name;  
    private String idNum;  
    private int age;  
  
    public int getAge() {  
        return age;  
    }  
  
    public String getName() {  
        return name;  
    }  
  
    public String getIdNum() {  
        return idNum;  
    }  
  
    public void setAge( int newAge) {  
        age = newAge;  
    }  
  
    public void setName(String newName) {  
        name = newName;  
    }  
  
    public void setIdNum( String newId) {  
        idNum = newId;  
    }  
}
```



Object Oriented Principles - Encapsulation

```
/* File name : RunEncap.java */  
public class RunEncap {  
  
    public static void main(String args[]) {  
        EncapTest encap = new EncapTest();  
        encap.setName("James");  
        encap.setAge(20);  
        encap.setIdNum("12343ms");  
  
        System.out.print("Name : " + encap.getName() + " Age : " + encap.getAge());  
    }  
}
```



Object Oriented Principles - Class and Objects

- A class is a user defined blueprint or prototype from which objects are created.
- Object : It is a basic unit of Object Oriented Programming and represents the real life entities. A typical Java program creates many objects, which as you know, interact by invoking methods.



