Object Oriented Programming

ICT2122

Encapsulation

P.H.P. Nuwan Laksiri
Department of ICT
Faculty of Technology
University of Ruhuna

Lesson 03 - OOP Concepts - Part 04

Recap

- Abstraction
- Abstraction in JAVA
- Abstract Methods
- Abstract Classes
- Hands-On

Outline

- Encapsulation
- Encapsulation – Hands-On
- Encapsulation - Advantages
- Abstraction vs Encapsulation

Object Oriented Concepts

- Object Oriented Programming simplifies the software development and maintenance by providing some concepts,
  - Object
  - Class
  - Inheritance
  - Polymorphism
  - Abstraction
  - Encapsulation

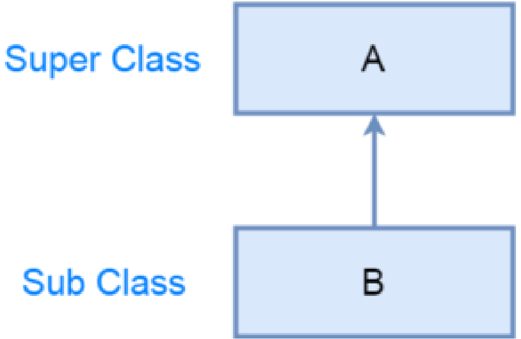
Classes and Objects

A class is like a cookie cutter; it defines the shape of objects

Objects are like cookies; they are instances of the class

Photograph courtesy of Guillaume Brialon on Flickr.

Inheritance



- Inheritance is a mechanism that allows
  - a subclass to inherit the properties and behaviors of a superclass.
- This means that the
  - subclass can access and use all the methods and variables of the superclass,
  - as well as add its own methods and variables.
- The subclass can also
  - override methods from the superclass to provide its own implementation.
- Inheritance enables
  - code reuse and makes it easier to manage and maintain complex systems
  - by reducing duplication and
  - providing a hierarchical structure for classes.
- It is a key feature of object-oriented programming and is widely used in Java

Polymorphism

- Poly-Morphism-> ability to have multiple forms (shapes) of the same thing.

- Polymorphism is the capability of an action or method to do different things based on the object that it is acting upon.

Abstraction

- Abstraction in Java is a mechanism that helps to reduce the complexity of a system by **hiding its implementation details from the user**.
- This means that the user only sees what is necessary to perform a certain task and does not need to know about the underlying implementation.

- For example,
  - sending a SMS, you just type the text and send the message.
  - You don't know the internal processing about the message delivery.

Encapsulation

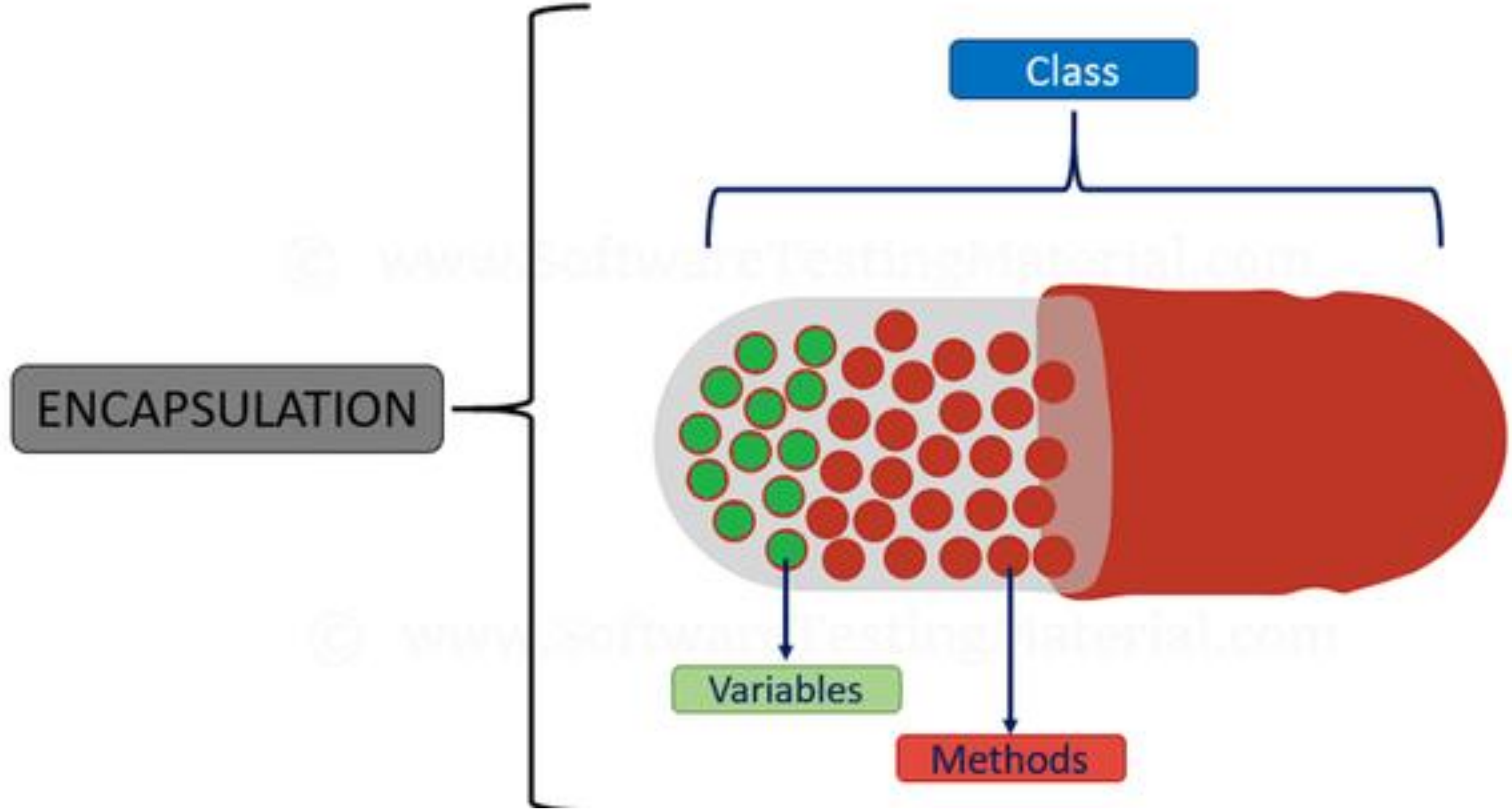
- Encapsulation in java is a **process of wrapping code and data together into a single unit.**

  example :

  capsule

  mixed of several medicines.

Encapsulation - Example



Encapsulation

- Encapsulation is the technique of **making the fields in a class private and providing access to the fields via public methods**.

- If a field is declared private, it cannot be accessed by any one outside the class, thereby hiding the fields within the class.

- For this reason, encapsulation is **also referred to as data(information) hiding.**

Encapsulation

- The **get methods** that allow a field to be viewed are known as **accessor** methods.
- The **set methods** that allow a field to be changed are known as **mutator** methods.

Encapsulation – Hands-On

```
public class MyEncapsulator
{
        //Private fimember, accessible only within the class
        private String name;

        //public setter (mutator) method to set the value
        public void setName(String name){
                this.name = name;
        }

        //public getter (accessor) method to get the value
        public String getName(){
                return name;
        }

}
```

Encapsulation – Try it

- The following *SalesPerson* class demonstrates encapsulation.
- Each of its fields is marked private, and there are public methods to access the fields.

Encapsulation – Try it

```
public class SalesPerson
{
        private String name;
        private int id;
        private float commissionRate;
        private double sales;

    SalesPerson(String name,int id, double commissionRate)
        {
                setName(name);
                this.id = id;
                setCommissionRate(commissionRate);
        }
```

Encapsulation - Try it

```
public void setName(String n)
{
        name = n;
}

public String getName()
{
        return name;
 }
```

Encapsulation - Try it

```
public void setCommissionRate(double newRate)
{
      if (newRate >= 0.0 && newRate <= 0.20)
       {
                 commissionRate = (float)newRate;
        }
       else
       {
                 System.out.println(“Rate must be between 0 and 20% ”);
        }
}
```

Encapsulation - Try it

```
public double getCommisssionRate()
{
        return commissionRate
}

public int getId()
{
        return id;
}

public void addToSales(double s)
{
        sales += s
}
```

Encapsulation - Try it

```
public double computeCommission()
{
        double commission = 0.0
        if (sales > 0.0)
        {
                 commission = sales * commissionRate;
         }
                 sales = 0.0; // start over
                 return commission;
    }
}
```

Encapsulation - Advantages

- By providing only setter or getter method, you can make the class read-only or write-only.
- It provides you the control over the data.
  - Suppose you want to set the value of id i.e. greater than 100 only, you can write the logic inside the setter method.
- A class can have total control over what is stored in its fields. The "SalesPerson" class demonstrates this with the "commissionRate" field, which can only be a value between 0.0 and 0.20
- The users of a class do not know how the class stores its data.
  - A class change the data type of a field, and users of the class do not need to change any of their code.

Abstraction vs. Encapsulation

- Often encapsulation is misunderstood with Abstraction.
  - **Encapsulation** is more about "**How**" to achieve a functionality
  - **Abstraction** is more about "**What**" a class can do.

- A simple example to understand this difference is a mobile phone.
  - Where the complex logic in the circuit board is encapsulated in a touch screen, and the user interface is provided to abstract it out.

Abstraction vs. Encapsulation

Abstraction

- Abstraction solves the issues at the design level.
- Abstraction is about hiding unwanted details while showing most essential information.
- Abstraction allows focusing on what the information object must contain

Encapsulation

- Encapsulation solves it implementation level.
- Encapsulation means hiding the code and data into a single unit.
- Encapsulation means hiding the internal details or mechanics of how an object does something for security reasons.

Summary

- Encapsulation
- Encapsulation – Hands-On
- Encapsulation - Advantages
- Abstraction vs Encapsulation

References

- How To Program (Early Objects)
  - By H .Deitel and P. Deitel
- Headfirst Java
  - By Kathy Sierra and Bert Bates

Questions ???

Thank You