OOPS (JAVA)
Lec-11

Saif Nalband

Contents

- Classes & Objects
- Encapsulation
- Constructors

Classes

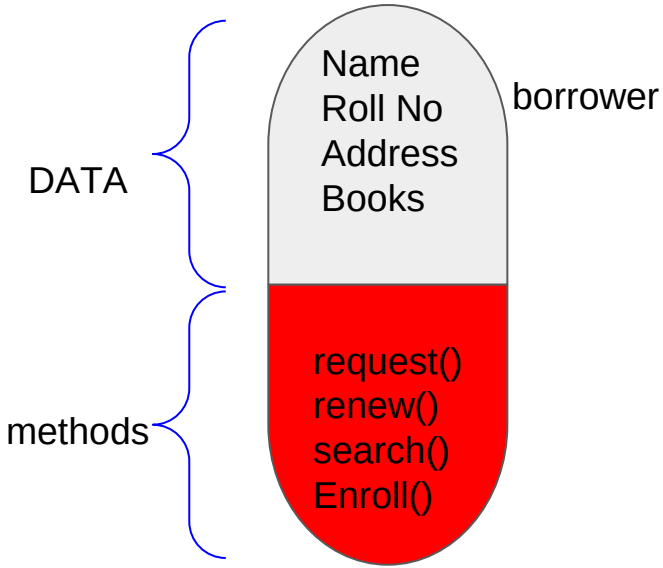
- The object is the core of JAVA programing.
- JAVA provides the concept of class to build objects.
- A class defines the shape and working of an object
- Class is logical while object is real.

What is a Class ?

- A class is group of objects, which have common properties.
- Its is a template or blueprint from which objects are created
- It is a logical entity

A class in JAVA can contain :

- Data
- Methods
- Constructors
- Blocks
- Nested Class(es) and interfaces



General Structure of a Class

```
class <class-name>{
                    <type> <variable>;
                    <type> <variable>;
                    ….
                    <type> <variable>;
                    <type> <method>(<parameters>);
                    <type> <method>(<parameters>);
                    …
                    <type> <method>(<parameters>);
}
```

encapsulation means data and operation are to be put together.

JAVA Class -- An Example

```
class circle {
        double x, y; // the coordinates of the circle
        double r; // The radius
}
```

Adding methods to Circle class : Demo110

```
class circle{
    double x,y;
    double r;
    // methods
    double circumferences(){
        return 2* 3.14159 * r;
    }
    // methods that return area
    double area(){
        return(22/7)* r * r;
    }
}
```

```
class Demo110{
    public static void main(String
args[]){
        Circle c = new Circle();
        c.x = 0.0;
        c.y = 0.0;
        c.r = 5.0;
System.out.println("Circumference " +
c.circumferences());
        System.out.println("Area " +
c.area());
    }
}
```

Multiple Object declarations : Demo111

```
class Demo111{
   public static void main(String args[]){
        Circle c1 = new Circle();
        Circle c2 = new Circle();
        // Initialize the circles
        c1.x = 3.0;
        c1.y = 4.0;
        c1.r = 5.0;
        c2.x = -4.0;
        c2.y = -8.0;
        c2.r = 10.0;
        System.out.println("Circumference Circle 1" + c1.circumference());
        System.out.println("Area Circle 1" + c1.area());
        System.out.println("Circumference Circle 2" + c2.circumference());
        System.out.println("Area Circle 2" + c2.area());
   }
}
```

Multiple Classes in a program

```
class circle{
    double x,y;
    double r;
    // methods
    double circumferences(){
        return 2* 3.14159 * r;
    }
    // mthods that return area
    double area(){
        return(22/7)* r * r;
    }
}
```

```
class Box{
  double width;
  double height;
  double depth;
  double area(){
     double a;
      a = (width*height +
height*depth + width*depth) * 2;
      return a;
  }
  double volume(){
      double v;
      v = width*height*depth;
      return v;
   }
}
```

Important notes

1. There should be a class which contains method main(). This class is called **main class**.
2. There should be only **one main class**.
3. The **name of the program file** should be same as the **name of the class** followed by **.java** as extension
4. If there is no main class, then there should be compilation error- **Run Time**

Java program without main class

DEMO113

Java program without main class -- Demo113

```
class circle{
    double x,y;
    double r;
    // methods
    double circumferences(){
        return 2* 3.14159 * r;
    }
    // mthods that return area
    double area(){
        return(22/7)* r * r;
    }
}
```

```
class Box{
  double width;
  double height;
  double depth;
  double area(){
     double a;
      a = (width*height +
height*depth + width*depth) * 2;
      return a;
  }
  double volume(){
      double v;
      v = width*height*depth;
      return v;
   }
}
```

Methods with parameters Demo114

```
class Circle {
   double x,y;
   double r;
   double circumference(){
       return 2*3.14159*r;
   }
   double area(){
       return (22/7)*r*r;
   }
   void setCircle(double a, double b,
double c){
       x = a; // Set center x-coordinate
       y = b; // Set center y-coordinate
       r = c; // Set radius
   }
}
```

```
class Demo114 {
   public static void main(String[]
args){
       Circle c1 = new Circle();
       Circle c2 = new Circle();
       // Initilise the circle
       c1.setCircle(1.0, 4.0, 5.0);
       c2.setCircle(3.0, 4.0, 10.0);
       System.out.println("Circumference
Circle 1" + c1.circumference());
       System.out.println("Area Circle
1" + c1.area());
       System.out.println("Circumference
Circle 2" + c2.circumference());
       System.out.println("Area Circle
2" + c2.area());
   }
}
```

Thank you and
Stay Home and Stay safe