Lab Sheet 08

Exercise 01:

```
abstract class BankAccount
{
	private int accountNumber;
	private double balance;
	public void setAccountNumber(int acccountNumber)
	{
		this.accountNumber=accountNumber;
	}
	public void setBalance(float balance)
	{
		this.balance=balance;
	}
	public int getAccountNumber()
	{
		return accountNumber;
	}
	public double getBalance()
	{
		return balance;
	}
	abstract double calculateInterest();
}
```

```
public class SavingsAccount extends BankAccount
{
	final double savingInterest=0.12;
	@Override
	double calculateInterest()
	{
		double balance=getBalance();
		return balance*savingInterest;
	}
}
public class CheckingAccount extends BankAccount
{
	final double checkingInterest=0.02;
	@Override
	double calculateInterest()
	{
		double balance=getBalance();
		return balance*checkingInterest;
	}
}
```

```
public class AccountObj
{
	public static void main(String[] args)
	{
		SavingsAccount sa=new SavingsAccount();
		sa.setAccountNumber(123);
		sa.setBalance(20000000);
		System.out.println("Savings Account Number: "+sa.getAccountNumber());
		System.out.println("Savings Account balance: "+sa.getBalance());
		double balance=sa.calculateInterest();
		System.out.println("Interest: "+balance+" \n");

		CheckingAccount ca=new CheckingAccount();
		ca.setAccountNumber(456);
		ca.setBalance(1000000);
		System.out.println("Savings Account Number: "+ca.getAccountNumber());
		System.out.println("Savings Account balance: "+ca.getBalance());
		double balance1=ca.calculateInterest();
		System.out.println("Interest: "+balance1);
	}
}
```

Exercise 02:

```
interface Shape
{
	abstract double calculateArea();
	abstract double calculatePerimeter();
}
public class Circle implements Shape
{
	private double radius;
	public Circle(double radius)
	{
		this.radius=radius;
	}
	public void setRadius(double radius)
	{
		this.radius=radius;
	}
	public double getRadius()
	{
		return radius;
	}
	@Override
	public double calculateArea()
	{
		return Math.PI*radius*radius;
	}
```

```
	@Override
	public double calculatePerimeter()
	{
		return 2*Math.PI*radius;
	}
}
public class Rectangle implements Shape
{
	private double width,height;
	public Rectangle(double width,double height)
	{
		this.width=width;
		this.height=height;
	}
	public void setData(double width,double height)
	{
		this.width=width;
		this.height=height;
	}
	public double getWidth()
	{
		return width;
	}
	public double getHeight()
	{
		return height;
	}
```

```
	@Override
	public double calculateArea()
	{
		return width*height;
	}
	@Override
	public double calculatePerimeter()
	{
		return (2*width+2*height);
	}
}
public class Triangle implements Shape
{
	private double base,height,side;
	public Triangle(double base,double height,double side)
	{
		this.base=base;
		this.height=height;
		this.side=side;
	}
	public void setData()
	{
		this.base=base;
		this.height=height;
		this.side=side;
	}
```

```
	public double getBase()
	{
		return base;
	}
	public double getHeight()
	{
		return height;
	}
	public double getSide()
	{
		return side;
	}
	@Override
	public double calculateArea()
	{
		return 0.5*base*height;
	}
	@Override
	public double calculatePerimeter()
	{
		return base+height+side;
	}
}
```

```
public class ShapeInterface
{
	public static void main(String[] args)
	{
		Circle c=new Circle(7);
		c.getRadius();
		System.out.println("Area is: "+c.calculateArea());
		System.out.println("Perimeter is: "+c.calculatePerimeter()+"'\n");

		Rectangle r=new Rectangle(10,5);
		r.getWidth();
		r.getHeight();
		System.out.println("Area is: "+r.calculateArea());
		System.out.println("Perimeter is: "+r.calculatePerimeter()+"'\n");

		Triangle t=new Triangle(10,5,6);
		t.getBase();
		t.getHeight();
		t.getSide();
		System.out.println("Area is: "+t.calculateArea());
		System.out.println("Perimeter is: "+t.calculatePerimeter()+"'\n");
	}
}
```