STT-NF

Object Oriented Programming

#4 Object and Class

Hilmy A. Tawakal & Agung Prayoga

October 3, 2017

Table of contents

Designing class

BankAccount.java

- Property ⇒ Variable
  - Account number → String
  - Account name → String
  - Balance → double
- Method ⇒ Function
  - Withdraw
    - return value → void (no return value)
    - arguments → amounts (double)
  - Deposit
    - return value → void (no return value)
    - arguments → amounts (double)
  - Check balance
    - return value → balance (double)
    - arguments → no arguments
  - Transfer

Creating class

BankAccount.java

```
public class BankAccount{
    private double balance;
    public BankAccount(){
            balance=0;
    }
    public double getBalance(){
            return balance;
    }
    public void deposit(double amount){
            balance=balance+amount;
    }
}
```

Property

BankAccount.java

```
public class BankAccount{
  /* Defining class property */
  private double balance;
  . . .
}
```

Method

BankAccount.java

```
public class BankAccount{
    /**
      method getBalance, without argument return double
    */
    public double getBalance(){
            return balance;
    }
    /**
      method deposit, one argument (double) without return (void)
    */
    public void deposit(double amount){
            balance=balance+amount;
    }
}
```

Constructor

BankAccount.java

```
public class BankAccount{
    /**
      method constructor, automatically called when creating
          instance
    */
    public BankAccount(){
            balance=0;
    }
}
```

Instantiation (create instance)

DemoBankAccount.java

```
public class DemoBankAccount{
    public static void main(String ar[]){
        /* create instance */
        BankAccount b1 = new BankAccount();
        BankAccount b2 = new BankAccount();
        BankAccount b3 = b1;
    }
}
```

Access method from object

DemoBankAccount.java

```
public class DemoBankAccount{
    public static void main(String ar[]){
        /* create instance */
        BankAccount b1 = new BankAccount();
        BankAccount b2 = new BankAccount();
        BankAccount b3 = b1;

        /* call method from b1 */
        double saldo = b1.getBalance();
        System.out.println("saldo b1:"+saldo);
        /* call method from b2 */
        b2.deposit(100.50);
        System.out.println("saldo b1:"+b1.getBalance()); // ?
        System.out.println("saldo b2:"+b2.getBalance()); // ?
    }
}
```

Object reference

DemoBankAccount.java

```
public class DemoBankAccount{
    public static void main(String ar[]){
        /* create instance */
        BankAccount b1 = new BankAccount();
        BankAccount b2 = new BankAccount();
        BankAccount b3 = b1;

        b1.deposit(99.99);
        b2.deposit(100.50);
        b3.deposit(45.765);

        System.out.println("saldo b1:"+b1.getBalance()); // ?
        System.out.println("saldo b2:"+b2.getBalance()); // ?
        System.out.println("saldo b3:"+b3.getBalance()); // ?
    }
}
```

Overloading method

Overloading → two or more methods with same name but different argument

```
public class BankAccount{
    public void deposit(double amount){
            balance=balance+amount;
    }
    public void deposit(int amount){
            balance=balance+amount;
    }
    public void deposit(int amount,double tax){
            balance= balance + amount - tax;
    }
}
```

Quiz1

1. Add property account name to class BankAccount
2. Modify constructor to initialize account name
3. Create method to print detail account (Account name and balance)
4. Overload constructor → without argument, and with string argument (account name)
5. Add method transfer → void, 2 argument → amount(double), dest (BankAccount)