Abstraction

Shristi Technology Labs

Contents

- What is Abstraction?
- Abstract classes and methods
- Final class, method & variable

What is Abstraction?

- Hiding the unwanted details.
- For future extensions

- Achieved using abstract classes and interfaces.

Abstract Class

- A class with atleast one abstract method
- Cannot be instantiated. (ie) can’t create objects
- Must have abstract keyword.
- Can also have normal methods
- Must be extended.
- Subclasses provide the functionality.

```
abstract  class Employee{
    // can have both abstract and normal methods
 }
```

Abstract Method

- A method with declaration and no definitions
- Must use abstract keyword
- Must be implemented in the sub classes.

abstract void calcSalary(int x);

Example

```
abstract class ABank{

    abstract void carLoan();
    abstract void housingLoan();

    void admin(){
          println("normal
    method");
    }
}
```

```
class Branch1 extends ABank{

    void carLoan(){
          println("car loan:15%");
    }
    void housingLoan(){
       println("Housing
    loan:20%");
    }
    void myMethod(){
          print(“ tax free“);
    }
}
```

Contd...

```
Class Branch2 extends ABank
abstract class Branch2 extends
    ABank{
 void carLoan(){
          println("car loan:10%");
    }
  void brDetails(){
          println("low interest:");
    }
}
```

housingLoan() not implemented. so make the class abstract and extend it.

```
class Subbranch extends
    Branch2{

    void housingLoan(){
          println("vehicle
    loan:22%");
    }
    int subPay(){
          println("int free
    loans:");
    return 1000;
    }
}
```

Subclass implements the method.

Contd...

```
class AbstractBank{


public static void main(String
    args[]){
    ABank ref;
     ref=new Branch1();
    ref.carLoan();
    ref.housingLoan();


    ref.admin();
    Branch1 br = new Branch1();
    br.myMethod();
```

```
 ref=new  SubBranch();
    ref.carLoan();
    ref.housingLoan();


Branch2 y;
    y=new Subbranch();
    y.brDetails();

Subbranch s=new
    Subbranch();
System.out.println(s.subPay())
    ;
}
}
```

More on abstraction

- Using the super class reference all the overridden methods and the methods of the super class can be called.

- To call the own methods of the sub class create objects of the sub class and call them

- When you use

  super-class ref = sub class object()

and call the overridden method only the **sub class**

method will be called

Final class

- Cannot be extended.
- Uses final keyword
- Some inbuilt classes are final.

(eg.) String, Math

```
public  final class Details{  }
```

Final Method

- Cannot be overridden by the subclasses
- Method must use final keyword

```
final void adminDetails(){
        print("confidential");
}
```

Final variable

- Are constants.
- Must be initialized.
- Mostly public to allow external access.
- By convention - variable name has to be in uppercase, as it is constant

final int BONUS = 3000;

Final - a glance

Final Classes

- Cannot be extended.
- Some inbuilt classes are final. eg. String, Math

eg. public final class Details{}

Final Methods

- Cannot be overridden.

eg. final int adminDetails(){ }

Final Variables

- Are constants. Must be initialized.
- Mostly public to allow external access

  eg. final int BONUS_AMOUNT = 3000;

Summary

- What is Abstraction?
- Abstract classes and methods
- Example
- Final class, method and Variable

Thank You