

26 October 2020

Aayush Shah

19BCE245

Practical 8

OOP Lab

Practical 8 A

Create an abstract class Instrument which is having the abstract function play. Create three more sub classes from Instrument which is Piano, Flute and Guitar. Override the play method inside all three classes printing a message. "Piano is playing tan tan tan tan" for Piano class. "Flute is playing toot toot toot toot" for Flute class. "Guitar is playing tin tin tin tin" for Guitar class. You must not allow the user to declare an object of Instrument class. Create an array of 10 Instruments. Assign different type of instrument to Instrument reference. Check for the polymorphic behavior of play method. Use the instanceof operator to print that which object stored at which index of instrument array.

CODE

```
abstract class Instrument {
    abstract void play();
}
class Piano extends Instrument{
    void play(){
        System.out.println("Piano is playing tan tan tan tan ");
    }
}
class Flute extends Instrument{
    void play(){
        System.out.println("Flute is playing toot toot toot toot ");
    }
}
class Guitar extends Instrument{
    void play(){
        System.out.println("Guitar is playing tin tin tin tin ");
    }
}
```

```

    }
}
public class Main{
    public static void main(String[] args) {
        Instrument[] arr = new Instrument[10];

        for (int i=0;i<10;i++) {
            switch (i%3) {
                case 0:
                    arr[i] = new Piano();
                    break;
                case 1:
                    arr[i] = new Flute();
                    break;
                default:
                    arr[i] = new Guitar();
                    break;
            }
        }

        for(int i=0;i<10;i++){
            if(arr[i] instanceof Piano)
                System.out.println("At index " + i + ", object of Piano class is stored");
            else if(arr[i] instanceof Flute)
                System.out.println("At index " + i + ", object of Flute class is stored");
            else if(arr[i] instanceof Guitar)
                System.out.println("At index " + i + ", object of Guitar class is
stored");
            else {
                System.out.println("none");
            }
        }
    }
}


```

INPUT :


-

OUTPUT :

```
At index 0, object of Piano class is stored
At index 1, object of Flute class is stored
At index 2, object of Guitar class is stored
At index 3, object of Piano class is stored
At index 4, object of Flute class is stored
At index 5, object of Guitar class is stored
At index 6, object of Piano class is stored
At index 7, object of Flute class is stored
At index 8, object of Guitar class is stored
At index 9, object of Piano class is stored
```

 Run Succeeded

Time 115 ms

 Instrument

Tabs: 4

50 lines, 1149 characters

CONCLUSION :

From the practical 8 A, We learned about the inbuilt java class *random* through which we can generate the random number of our choice and *use of instanced keyword* through which we can identify that whether the object is the instance of the given class or not. Also we implemented the concept of inheritance in this source code.

Practical 8 B

Create an abstract class *Compartment* to represent a rail coach.

Provide an abstract function *notice* in this class. Derive *FirstClass*, *Ladies*, *General* and *Luggage* classes from the *compartment* class. Override the *notice* function in each of them to print notice suitable to the type of the compartment. Create a class *TestCompartment*. Write main function to do the following: Declare an array of *Compartment* pointers of size 10. Create a compartment of a type as decided by a randomly generated integer in the range 1 to 4. Check the polymorphic behavior of the *notice* method.

CODE

```
import java.util.*;

abstract class Compartment {
    abstract void notice();
}

class FirstClass extends Compartment{
    void notice(){
        System.out.println("Inside notice function of FirstClass");
    }
}

class Ladies extends Compartment{
    void notice(){
        System.out.println("Inside notice function of Ladies");
    }
}

class General extends Compartment{
    void notice(){
        System.out.println("Inside notice function of General");
    }
}

class Luggage extends Compartment{
    void notice(){
        System.out.println("Inside notice function of Luggage");
    }
}

public class TestCompartment{
    public static void main(String[] args) {
        Random rand = new Random();
        int randomNumber;
        Compartment[] c = new Compartment[10];
        for(int i=0;i<10;i++){
            randomNumber = rand.nextInt(4)+1;
            System.out.println("random number generated for index " + i + "
is " + randomNumber);
            switch (randomNumber) {
                case 1:
                    c[i] = new FirstClass();
                    break;
                case 2:
                    c[i] = new Ladies();
                    break;
                case 3:
                    c[i] = new General();
                    break;
                case 4:
                    c[i] = new Luggage();
```

```
                break;
            default:
                System.out.println("NONE");
                break;
        }
    }
    for (int i=0;i<10;i++) {
        c[i].notice();
    }
}
```

INPUT :

-

OUTPUT :

```
random number generated for index 0 is 2
random number generated for index 1 is 3
random number generated for index 2 is 3
random number generated for index 3 is 4
random number generated for index 4 is 3
random number generated for index 5 is 2
random number generated for index 6 is 2
random number generated for index 7 is 1
random number generated for index 8 is 4
random number generated for index 9 is 1
Inside notice function of Ladies
Inside notice function of General
Inside notice function of General
Inside notice function of Luggage
Inside notice function of General
Inside notice function of Ladies
Inside notice function of Ladies
Inside notice function of FirstClass
Inside notice function of Luggage
Inside notice function of FirstClass
```

✓ Run Succeeded

Time 127 ms

Symbol ↕

Tabs: 4 ↕

Line 12, Column 2

CONCLUSION :

From the practical 8 B, We again got a hands-on Inheritance and the random class. Here we used the random class to generate the random numbers of our choice i.e. from 1 to 4. We can

also use Math.random to generate the random numbers. Also we used the concepts of abstract class through which we cannot instantiated.

Practical 8 C

Consider an example of declaring the examination result. Design three classes: Student, Exam, and Result. The Student class has data members such as those representing roll number, name etc. Create the class Exam by inheriting the Student class. The Exam class adds fields representing the marks scored in six subjects. Derive the Result from the Exam class and it has its own fields such as total_Marks. Write an interactive program to model this relationship.

CODE :

```
import java.util.*;

abstract class Student {
    int rollNumber;
    String name;
}
abstract class Exam extends Student{
    static int[] marks = new int[6];
}
class Result extends Exam{
    int totalMarks;
    int calculateTotalMarks(){
        for(int i=0;i<6;i++)
            totalMarks += super.marks[i];
        return this.totalMarks;
    }
}
public class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Student s1;
        Exam e1;
        Result r1 = new Result();
    }
}
```

```
s1 = r1;
e1 = r1;

System.out.print("Enter roll number : ");
s1.rollNumber = sc.nextInt();
System.out.print("Enter name : ");
s1.name = sc.next();

for(int i=0;i<6;i++){
    System.out.print("Enter marks for subject : " + (i+1) + " : ");
    e1.marks[i] = sc.nextInt();
}

System.out.println("Total marks : " + r1.calculateTotalMarks());
}
```

INPUT :

```
245
Aayush
50
60
70
80
90
100
```

OUTPUT :

```
Enter roll number : 245
Enter name : Aayush
Enter marks for subject : 1 : 50
Enter marks for subject : 2 : 60
Enter marks for subject : 3 : 70
Enter marks for subject : 4 : 80
Enter marks for subject : 5 : 90
Enter marks for subject : 6 : 100
Total marks : 450
```

✓ Run Succeeded Time 191 ms

M main ▾ Tabs: 4 ▾ Line 29, Column 38

CONCLUSION :

For the practical 8 C, We again used the concepts of inheritance and abstract class. We made Student and Exam as an abstract class. Exam is the child class of Student and Result is the child class of Exam. Among this, we created one object of Result class then used the parent class variable through super keyword and then calculated the total marks.

Practical 8 D

Create a class Medicine to represent a drug manufactured by a pharmaceutical company. Provide a function displayLabel() in this class to print Name and address of the company. Derive Tablet, Syrup and Ointment classes from the Medicine class. Override the displayLabel() function in each of these classes to print additional information suitable to the type of medicine. For example, in case of tablets, it could be “store in a cool dry place”, in case of ointments it could be “for external use only” etc. Create a class TestMedicine. Write main function to do the following: Declare an array of Medicine references of size 10 Create a medicine object of the type as decided by a randomly generated integer in the range 1 to 3. Check the polymorphic behavior of the displayLabel() method.

CODE :

```
import java.util.*;

class Medicine {
    String companyName;
    String companyAddress;

    void displayLabel(){
```



```
        System.out.println("Name of the company : " + this.companyName);
        System.out.println("Address of the company : " + this.companyAddress);
    }
}
class Tablet extends Medicine{
    void displayLabel(){
        super.displayLabel();
        System.out.println("Store in a cool dry place");
    }
}
class Syrup extends Medicine{
    void displayLabel(){
        super.displayLabel();
        System.out.println("Used to treat coughs");
    }
}
class Ointment extends Medicine{
    void displayLabel(){
        super.displayLabel();
        System.out.println("For external use only");
    }
}
public class TestMedicine{
    public static void main(String[] args) {
        Medicine arr[] = new Medicine[10];
        Random r = new Random();
        int randomNumber;

        for(int i=0;i<10;i++){
            randomNumber = r.nextInt(3) + 1;
            System.out.println("For index " + i + ", Random number
generated is : " + randomNumber);
            switch (randomNumber) {
                case 1:
                    arr[i] = new Tablet();
                    arr[i].companyName = new String("AAYUSH1");
                    arr[i].companyAddress = new String("Vadodara");
                    break;
                case 2:
                    arr[i] = new Syrup();
                    arr[i].companyName = new String("AAYUSH2");
                    arr[i].companyAddress = new String("Ahemdabad");
                    break;
                case 3:
                    arr[i] = new Ointment();
                    arr[i].companyName = new String("AAYUSH3");
                    arr[i].companyAddress = new String("Gandhinagar");
                    break;
                default:
```

```

        System.out.println("NONE");
        break;
    }
}
for (int i=0;i<10;i++) {
//      System.out.println("Name of the company : " +
arr[i].companyName);
//      System.out.println("Address of the company : " +
arr[i].companyAddress);
        arr[i].displayLabel();
    }
}
}

```

INPUT :




-

OUTPUT :

```

For index 0, Random number generated is : 1
For index 1, Random number generated is : 2
For index 2, Random number generated is : 2
For index 3, Random number generated is : 1
For index 4, Random number generated is : 1
For index 5, Random number generated is : 2
For index 6, Random number generated is : 2
For index 7, Random number generated is : 1
For index 8, Random number generated is : 2
For index 9, Random number generated is : 2
Name of the company : AAYUSH1
Address of the company : Vadodara
Store in a cool dry place
Name of the company : AAYUSH2
Address of the company : Ahemdabad
Used to treat coughs
Name of the company : AAYUSH2
Address of the company : Ahemdabad
Used to treat coughs
Name of the company : AAYUSH1
Address of the company : Vadodara
Store in a cool dry place
Name of the company : AAYUSH1
Address of the company : Vadodara
Store in a cool dry place
Name of the company : AAYUSH2
Address of the company : Ahemdabad
Used to treat coughs
Name of the company : AAYUSH2
Address of the company : Ahemdabad
Used to treat coughs

```

 Run Succeeded | Time 140 ms | Symbol  | Tabs: 4  | 68 lines, 1766 characters

CONCLUSION :

For the practical 8 D, We used the various concepts of java like inheritance , method overriding and use of random class. First we created a parent class named Medicine then we created Tablet,Syrup and Ointment classes which are child classes of Medicine. Then for testing we created main class named TestMedicine in which we generated random number using the inbuilt random class in java.

Practical 8 E

Create a class Car which contains members speed, noOfGear. The class has a method drive() which is responsible to provide starting speed and noOfGears to a Car. Implement display() method which will display all attributes of Car class. The class SportCar is derived from the class Car which adds new features AirBallonType. When this method is invoked, initial speed and gear status must be displayed on console. Override the display method which display all attribute of the SportCar. Make use of super class display() method.

CODE :

```
class Car {
    double speed = 0;
    int noOfGear = 5;
    void drive(){
        this.speed = 30;
        this.noOfGear = 5;
    }
    void display(){
        System.out.println("Speed of the car : " + this.speed);
        System.out.println("No. of Gear : " + this.noOfGear);
    }
}
class SportCar extends Car{
    double speed = 50;
```

```
double noOfGear = 6;
void AirBallonType(){
    speed = 70;
    noOfGear = 6;
    System.out.println("Initial speed : " + speed);
    System.out.println("No. of Gear : " + noOfGear);
}
void display(){
    super.display();
    System.out.println("Speed of the sports car : " + this.speed);
    System.out.println("No. of Gear for sports car : " + this.noOfGear);
}
}
public class Main{
    public static void main(String[] args) {
        Car c1 = new Car();
        System.out.println("Starting Speed : " + c1.speed);
        System.out.println("No. of gear : " + c1.noOfGear);
        c1.drive();
        c1.display();

        SportCar c2 = new SportCar();
        System.out.println("Starting Speed : " + c2.speed);
        System.out.println("No. of gear : " + c2.noOfGear);
        c2.AirBallonType();
        c2.display();
    }
}
```

INPUT :

-

OUTPUT :

```
Starting Speed : 0.0
No. of gear : 5
Speed of the car : 30.0
No. of Gear : 5
Starting Speed : 50.0
No. of gear : 6.0
Initial speed : 70.0
No. of Gear : 6.0
Speed of the car : 0.0
No. of Gear : 5
Speed of the sports car : 70.0
No. of Gear for sports car : 6.0
```

✓ Run Succeeded Time 147 ms Car ↕ Tabs: 4 ↕ 42 lines, 1054 characters

CONCLUSION :

For the practical 8 E, We made use of the concept inheritance in java through which we made one parent class Car and its child class SportCar. We also used super keyword for using parent class variables and methods.

Practical 8 F

A super class Record has been defined to store the names and ranks of 50 students. Define a sub class Rank to find the highest rank along with the name. The details of both classes are given below:

Class name : Record

Data Members / instance variables:

- name[] : to store the names of students
- rnk[] : to store the ranks of students

Member functions:

- Record() : constructor to initialize data members
- void readvalues() : to store names and ranks
- void display() : displays the names and the corresponding ranks

Class name : Rank

Data Members / instance variables:

- index : integer to store the index of the topmost rank

Member functions

- Rank() : constructor to invoke the base class constructor and to initialize index to 0.
- void highest() : finds the index location of the topmost rank and stores it in index without sorting the array
- void display() : displays the name and ranks along with the name having the topmost rank.

Specify the class Record giving details of the constructor(), void readvalues(), void display(). Using the concept of inheritance, specify the class Rank giving details of constructor(), void highest() and void display().

CODE :

```
import java.util.Scanner;
import javax.net.ssl.*;

class Record {
```

```

Scanner sc = new Scanner(System.in);
static String[] name = new String[50];
static int[] rnk = new int[50];
int index;

Record(){
    index = 0;
}
void readValues(){
    for(int i=0;i<50;i++){
        System.out.print("Enter name of Student " + (i+1) + " : ");
        this.name[i] = sc.next();
        System.out.print("Enter rank of Student " + (i+1) + " : ");
        this.rnk[i] = sc.nextInt();
    }
}
void display(){
    for(int i=0;i<50;i++){
        System.out.println("Name of Student " + (i+1) + " : " + name[i]);
        System.out.println("rank of Student " + (i+1) + " : " + rnk[i]);
        System.out.println();
    }
}
}

class Rank extends Record{

    Rank(){
        super();
    }

    int highest(){
        int highestRank = rnk[0];
        for(int i=0;i<50;i++){
            if(highestRank<rnk[i]){
                highestRank = rnk[i];
                super.index = i;
            }
            else {
                continue;
            }
        }
        return super.index;
    }
    void display(int index){
        System.out.println("Detail of student who has highest rank : ");
        System.out.println("Name of Student : " + name[this.index]);
        System.out.println("rank of Student : " + rnk[this.index]);
    }
}

```

```
}  
public class Main{  
    public static void main(String[] args) {  
        Record r1 = new Record();  
        r1.readValues();  
        r1.display();  
        Rank R1 = new Rank();  
        R1.display(R1.highest());  
    }  
}
```

INPUT :

Giving 50 inputs for each students which includes name and rank

Sample OUTPUT (for 5 students) :

```
Enter name of Student 1 : q  
Enter rank of Student 1 : 1  
Enter name of Student 2 : w  
Enter rank of Student 2 : 3  
Enter name of Student 3 : e  
Enter rank of Student 3 : 5  
Enter name of Student 4 : r  
Enter rank of Student 4 : 2  
Enter name of Student 5 : t  
Enter rank of Student 5 : 4  
Name of Student 1 : q  
rank of Student 1 : 1  
  
Name of Student 2 : w  
rank of Student 2 : 3  
  
Name of Student 3 : e  
rank of Student 3 : 5  
  
Name of Student 4 : r  
rank of Student 4 : 2  
  
Name of Student 5 : t  
rank of Student 5 : 4  
  
Detail of student who has highest rank :  
Name of Student : e  
rank of Student : 5
```

Run Succeeded Time 238 ms readValues Tabs: 4 Line 14, Column 24

CONCLUSION :

For the practical 8 E, We made use of the concept inheritance in java through which we made one parent class Car and its child class SportCar. We also used super keyword for using parent class variables and methods.