

JAVA-OOPS
ISHWARI RAJMOHAN
230701118
CSE B

WEEK 7

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation



Show one page at a time

Finish review

Status	Finished
Started	Tuesday, 1 October 2024, 8:08 PM
Completed	Tuesday, 1 October 2024, 8:11 PM
Duration	3 mins 33 secs

Question **1**

Correct

Marked out of 5.00

Flag question

create an interface Playable with a method play() that takes no arguments and returns void. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play() method to play the respective sports.

```
interface Playable {
    void play();
}

class Football implements Playable {
    String name;
    public Football(String name){
        this.name=name;
    }
    public void play() {
        System.out.println(name+" is Playing football");
    }
}
```

Similarly, create Volleyball and Basketball classes.

Sample output:

Sadhvin is Playing football
Sanjay is Playing volleyball
Sruthi is Playing basketball

For example:

Test	Input	Result
1	Sadhvin	Sadhvin is Playing football
	Sanjay	Sanjay is Playing volleyball
	Sruthi	Sruthi is Playing basketball
2	Vijay	Vijay is Playing football
	Arun	Arun is Playing volleyball
	Balaji	Balaji is Playing basketball

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2
3
4
5 interface Playable {
6
7     void play();
8
9 }
10
11
12
13 class Football implements Playable {
14
15     String name;
16
17
18
19     public Football(String name) {
20
21         this.name = name;
22
23     }
24
25
26
27     @Override
28
29     public void play() {
30
31         System.out.println(name + " is Playing football");
32
33     }
34
35 }
36
37
38
39 class Volleyball implements Playable {
40
41     String name;
42
43
44
45     public Volleyball(String name) {
46
47         this.name = name;
48
49     }
50
51
52 }
```

```
53     @Override
54
55     public void play() {
56
57         System.out.println(name + " is Playing volleyball");
58
59     }
60
61 }
62
63
64
65 class Basketball implements Playable {
66
67     String name;
68
69
70
71     public Basketball(String name) {
72
73         this.name = name;
74
75     }
76
77
78
79     @Override
80
81     public void play() {
82
83         System.out.println(name + " is Playing basketball");
84
85     }
86
87 }
88
89
90
91 public class Main {
92
93     public static void main(String[] args) {
94
95         Scanner scanner = new Scanner(System.in);
96
97
98
99         String footballPlayerName = scanner.nextLine();
100
101         String volleyballPlayerName = scanner.nextLine();
102
103         String basketballPlayerName = scanner.nextLine();
104
```

```

104
105
106
107     Playable footballPlayer = new Football(footballPlayerName);
108
109     Playable volleyballPlayer = new Volleyball(volleyballPlayerName);
110
111     Playable basketballPlayer = new Basketball(basketballPlayerName);
112
113
114
115     footballPlayer.play();
116
117     volleyballPlayer.play();
118
119     basketballPlayer.play();
120
121
122
123     scanner.close();
124
125 }
126
127 }
128
129

```

	Test	Input	Expected	Got	
✓	1	Sadhvin Sanjay Sruthi	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	✓
✓	2	Vijay Arun Balaji	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	✓

Passed all tests! ✓

Question 2

Correct

Marked out of
5.00

Flag question

RBI issues all national banks to collect interest on all customer loans.

Create an RBI interface with a variable `String parentBank="RBI"` and abstract method `rateOfInterest()`.

RBI interface has two more methods default and static method.

default void `policyNote()` {

`System.out.println("RBI has a new Policy issued in 2023.");`

}

static void `regulations()`{

`System.out.println("RBI has updated new regulations on 2024.");`

}

Create two subclasses SBI and Karur which implements the RBI interface.

Provide the necessary code for the abstract method in two sub-classes.

Sample Input/Output:

RBI has a new Policy issued in 2023

RBI has updated new regulations in 2024.

SBI rate of interest: 7.6 per annum.

Karur rate of interest: 7.4 per annum.

For example:

Test	Result
1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.

Answer: (penalty regime: 0 %)

```
1 interface RBI {  
2  
3     String parentBank = "RBI";  
4  
5     double rateOfInterest();  
6  
7  
8  
9     default void policyNote() {  
10  
11         System.out.println("RBI has a new Policy issued in 2023");  
12  
13     }  
14  
15  
16  
17     static void regulations() {  
18  
19         System.out.println("RBI has updated new regulations in 2024.");  
20
```

```
21     }
22
23 }
24
25
26
27 class SBI implements RBI {
28
29     public double rateOfInterest() {
30
31         return 7.6;
32
33     }
34
35 }
36
37
38
39 class Karur implements RBI {
40
41     public double rateOfInterest() {
42
43         return 7.4;
44
45     }
46
47 }
48
49
50
51 public class Main {
52
53     public static void main(String[] args) {
54
55         SBI sbi = new SBI();
56
57         Karur karur = new Karur();
58
59
60
61         sbi.policyNote();
62
63         RBI.regulations();
64
65
66
67         System.out.println("SBI rate of interest: " + sbi.rateOfInterest() + " per annum.");
68
69         System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum.");
70
71     }
72 }
```

```
71 | }  
72 |  
73 | }  
74 |
```



	Test	Expected	Got	
✓	1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of
5.00

🚩 Flag question

Create interfaces shown below.

```
interface Sports {  
    public void setHomeTeam(String name);  
    public void setVisitingTeam(String name);  
}
```

```
interface Football extends Sports {  
    public void homeTeamScored(int points);  
    public void visitingTeamScored(int points);  
}
```

create a class College that implements the Football interface and provides the necessary functionality to the abstract methods.

sample Input:

Rajalakshmi

Saveetha

22

21

Output:

Rajalakshmi 22 scored

Saveetha 21 scored

Rajalakshmi is the Winner!

For example:

Test	Input	Result
1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!

```
55 import java.util.Scanner;
56
57
58
59 interface Sports {
60
61     void setHomeTeam(String name);
62
63     void setVisitingTeam(String name);
64
65 }
66
67
68
69 interface Football extends Sports {
70
71     void homeTeamScored(int points);
72
73     void visitingTeamScored(int points);
74
75 }
76
77
78
79 class College implements Football {
80
81     String homeTeam;
82
83     String visitingTeam;
84
85     int homeScore;
86
87     int visitingScore;
88
89
90
91     public void setHomeTeam(String name) {
92         |   this.homeTeam = name;
93     }
94
95
96
97
98
99     public void setVisitingTeam(String name) {
100         |   this.visitingTeam = name;
101     }
102
103 }
104
```

```
104
105
106
107 + public void homeTeamScored(int points) {
108     |     this.homeScore = points;
109
110 }
111
112
113
114
115 + public void visitingTeamScored(int points) {
116     |     this.visitingScore = points;
117
118 }
119
120
121
122
123 + public void displayResult() {
124     |
125     |     System.out.println(homeTeam + " " + homeScore + " scored");
126
127     |     System.out.println(visitingTeam + " " + visitingScore + " scored");
128
129
130
131 +     |     if (homeScore > visitingScore) {
132         |         System.out.println(homeTeam + " is the winner!");
133
134     |     } else if (visitingScore > homeScore) {
135 +         |         System.out.println(visitingTeam + " is the winner!");
136
137     |     } else {
138 +         |         System.out.println("It's a tie match.");
139
140     |     }
141
142 }
143
144
145 }
146
147 }
148
149
```

```

150
151 public class Main {
152
153     public static void main(String[] args) {
154
155         Scanner scanner = new Scanner(System.in);
156
157
158
159         College match = new College();
160
161
162
163         String homeTeam = scanner.nextLine();
164         String visitingTeam = scanner.nextLine();
165
166         int homeScore = scanner.nextInt();
167         int visitingScore = scanner.nextInt();
168
169
170
171
172
173         match.setHomeTeam(homeTeam);
174         match.setVisitingTeam(visitingTeam);
175
176         match.homeTeamScored(homeScore);
177
178         match.visitingTeamScored(visitingScore);
179
180
181
182
183         match.displayResult();
184
185
186
187         scanner.close();
188
189     }
190
191 }

```

	Test	Input	Expected	Got	
✓	1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	✓
✓	2	Anna Balaji 21 21	Anna 21 scored Balaji 21 scored It's a tie match.	Anna 21 scored Balaji 21 scored It's a tie match.	✓