

ობიექტზე ორიენტირებული დაპროგრამება და აპლიკაციები (Java-ს
ბაზაზე). II სემესტრი. 2020-2021 წელი
(შუალედური გამოცდის ნიმუში)

კლასების იერარქია

Question 1

მოცემულია კლასების იერარქია

```
class A {}  
class B extends A {}  
class C extends A {}
```

რომელი გამონათქვამებია ჭეშმარიტი?

Select one:

- a. B სუპერკლასია A კლასისათვის
- b. A სუპერკლასია B კლასისათვის
- c. C სუპერკლასია B კლასისათვის
- d. C სუპერკლასია A კლასისათვის

ცვლადები მეკვიდრობითობის დროს

Question 2

მოცემულია კლასების აღწერა

```
class A {  
    double a=1;  
    double b=5;  
    void sum(){  
        System.out.print(a+b);  
    }  
}  
class B extends A {  
    void mul(){  
        System.out.print(a*b);  
    }  
}
```

რომელი გამონათქვამია ჭეშმარიტი?

Select one:

- a. B კლასში a და b ცვლადები გამოცხადებული არაა, ამიტომ მათი გამოყენება დაუშვებელია
- b. B კლასში a და b ცვლადებს საწყისი მნიშვნელობები არ ენიჭებათ, ამიტომ შეცდომაა
- c. B კლასში a და b ცვლადების გამოყენება არ შეიძლება, ვინაიდან ასეთი ცვლადები სუპერკლასში უკვე გამოცხადებულია
- d. B კლასში a და b ცვლადების გამოყენება დასაშვებია, ვინაიდან ისინი სუპერკლასშია აღწერილი

მეთოდების გადატვირთვა/გადაფარვა მემკვიდრეობითობის დროს

Question 3

მოცემულია ორი კლასი:

```
class BaseClass {
    int meth(int i){
        return i;
    }
}
class SubClass extends BaseClass {
    int meth(int x) {
        return x*x;
    }
    void meth() {
        System.out.println("uparametro meTodi!");
    }
    int meth(int a, int b){
        return a+b;
    }
}
```

რომელი გამონათქვამებია ჭეშმარიტი?

Select one or more:

- a. ქვეკლასში void meth() მეთოდი სუპერკლასის მეთოდის გადატვირთვაა
- b. ქვეკლასის int meth(int x) მეთოდი სუპერკლასის მეთოდის გადატვირთვაა
- c. ქვეკლასში int meth(int a, int b) მეთოდი სუპერკლასის მეთოდის გადაფარვაა
- d. ქვეკლასში int meth(int a, int b) მეთოდი სუპერკლასის მეთოდის გადატვირთვაა
- e. ქვეკლასში void meth() მეთოდი სუპერკლასის მეთოდის გადაფარვაა
- f. ქვეკლასის int meth(int x) მეთოდი სუპერკლასის მეთოდის გადაფარვაა

super-ის გამოყენება კონსტრუქტორში

Question 4

რა დაიბეჭდება კონსოლზე პროგრამის შესრულების შედეგად?

```
class Base{
    Base(){
        System.out.print(" Base კლასის კონსტრუქტორი");
    }
}
class SubClass extends Base{
    SubClass(){
        super();
        System.out.print(" SubClass კლასის კონსტრუქტორი");
    }
}
class Test {
    public static void main (String args[]){
        SubClass ob = new SubClass();
    }
}
```

```
}
```

Select one:

- a. Base კლასის კონსტრუქტორი
- b. Base კლასის კონსტრუქტორი SubClass კლასის კონსტრუქტორი
- c. არაფერი არ დაიბეჭდება
- d. SubClass კლასის კონსტრუქტორი

super-ის გამოყენება მშობელი კლასის ცვლადზე წვდომისათვის

Question 5

რა დაიბეჭდება კონსოლზე პროგრამის შესრულების შედეგად?

```
class Base{  
int a=17;  
}  
class SubClass extends Base{  
int a=12;  
int meth(){  
return super.a;  
}  
}  
class Test {  
public static void main (String args[]){  
SubClass ob = new SubClass();  
System.out.print(ob.a);  
System.out.println(" "+ob.meth());  
}  
}
```

Select one:

- a. 17 12
- b. 12 12
- c. 12 17
- d. 17 17

კლასების ტიპების დაყვანა

Question 6

მოცემულია კლასების იერარქია

1. **class A** {}
2. **class B extends A** {}
3. **class C extends B** {}
4. **public class** Test {
5. **public static void** main(String[] args) {
6. B b = **new** C();
7. C c = **new** C();
8. A a = c;

9. a = b;
10. c = a;
11. }
12. }

რომელი გამონათქვამია სწორი?

Select one:

- a. Java მკაცრი ტიპიზაციის ენაა ამიტომ სხვადასხვა ტიპის ცვლადების მინიჭება დაუშვებელია
- b. 9 სტრიქონში შეცდომაა, ვინაიდან სუპერკლასის ტიპის ცვლადს ენიჭება ქვეკლასის ტიპის ობიექტი
- c. პროგრამაში შეცდომა არაა
- d. 10 სტრიქონში შეცდომაა, ვინაიდან ქვეკლასის ტიპის ცვლადს ენიჭება სუპერკლასის ტიპის ობიექტი

დინამიკური დისპეტერიზაცია

Question 7

მოცემულია კლასების იერარქია

```
class Base{  
    int meth(int x){  
        System.out.print(" "+ ++x);  
        return x;  
    }  
}  
class SubClass extends Base{  
    int meth(int y){  
        System.out.print(" "+y++);  
        return y;  
    }  
}  
class Test {  
    public static void main (String args[]){  
        SubClass s = new SubClass();  
        Base b = new Base();  
        Base ob;  
        ob = s;  
        System.out.print(" "+ob.meth(1));  
        ob = b;  
        System.out.println(" "+ob.meth(1));  
    }  
}
```

რა დაიბეჭდება?

Select one:

- a. 1 2 3 3
- b. 1 2 2 2
- c. 1 1 2 2
- d. 2 2 2 1

აბსტრაქტული მეთოდები

Question 8

მოცემულია კლასების იერარქია

```
1 abstract class Abstr {
2 abstract void meth(int i);
3 }
4 class B extends Abstr {
5 void meth(int x) {
6 System.out.println("B კლასის მეთოდი");
7 }
8 }
9 public class Test {
10 public static void main(String[] args) {
11 Abstr obj = new Abstr();
12 }
13 }
```

რომელი გამონათქვამია სწორი?

Select one:

- a. კომპილატორი შეცდომას არ მოგვცემს
- b. 5-ე სტრიქონში მეთოდის პარამეტრის სახელი არ ემთხვევა 2-ე სტრიქონის მეთოდის პარამეტრის სახელს, ამიტომ შეცდომაა
- c. 2-ე სტრიქონში მეთოდის ტანი არაა აღწერილი, ამიტომ კომპილატორი სინტაქსურ შეცდომას მოგვცემს
- d. 11-ე სტრიქონში შეცდომაა, ვინაიდან აბსტრაქტული ობიექტის შექმნის მცდელობაა

აბსტრაქტული მეთოდების რეალიზაცია

Question 9

მოცემულია კლასების იერარქია

```
1 abstract class Abstr {
2 abstract void meth(int x);
3 }
4 class B extends Abstr {
5 void meth() {
6 System.out.println("B კლასის მეთოდი");
7 }
8 }
9 public class Test {
10 public static void main(String[] args) {
11 B ob = new B();
12 }
13 }
```

რომელი გამონათქვამია სწორი?

Select one:

- a. კომპილატორი შეცდომას არ მოგვცემს

- b. კომპილატორი მოგვცემს შეცდომას, ვინაიდან 2-ე სტრიქონში აღწერილი აბსტრაქტული მეთოდის რეალიზაციას ქვეკლასი არ ახდენს
- c. 5-ე სტრიქონში მეთოდის სახელი არ შეიძლება ემთხვეოდეს სუპერკლასის 2-ე სტრიქონში მითითებულ მეთოდის სახელს
- d. კომპილატორი მოგვცემს შეცდომას, ვინაიდან სუპერკლასის 2-ე სტრიქონში მეთოდს ერთი პარამეტრი აქვს

final მოდიფიკატორი გადაფარვის აკრძალვისათვის

Question 10

მოცემულია პროგრამის ტექსტი

```
1 class A {  
2   final void meth() {  
3     System.out.println("A კლასის მეთოდი");  
4   }  
5 }  
6 class B extends A {  
7   void meth() {  
8     System.out.println("B კლასის მეთოდი");  
9   }  
10 }
```

რომელი გამონათქვამია სწორი?

Select one:

- a. კომპილატორი შეცდომას არ მოგვცემს
- b. სუპერკლასში და ქვეკლასში ერთნაირი სახელით მეთოდების გამოცხადება არ შეიძლება
- c. final მოდიფიკატორის მეთოდის წინ მითითება დაუშვებელია
- d. ვინაიდან 2-ე სტრიქონში სუპერკლასის მეთოდი გამოცხადებულია final მოდიფიკატორით, ამიტომ ქვეკლასში მისი გადაფარვა არ შეიძლება

final მოდიფიკატორი მემკვიდრეობითობის აკრძალვისათვის

Question 11

მოცემულია პროგრამის ტექსტი

```
1 final class A {  
2   void meth() {  
3     System.out.println("A კლასის მეთოდი");  
4   }  
5 }  
6 class B extends A {  
7   void meth() {  
8     System.out.println("B კლასის მეთოდი");  
9   }  
10 }
```

რომელი გამონათქვამია სწორი?

Select one:

- a. კომპილატორი შეცდომას არ მოგვცემს
- b. სუპერკლასში და ქვეკლასში ერთნაირი სახელით მეთოდების გამოცხადება არ შეიძლება
- c. final მოდიფიკატორის კლასის აღწერის წინ მითითება დაუშვებელია
- d. ვინაიდან 1-ლ სტრიქონში A კლასი აღწერილია final მოდიფიკატორით, ამიტომ მას არ შეიძლება ქვეკლასი ჰქონდეს

Object კლასის მეთოდები

Question 12

მონიშნეთ Object კლასის მეთოდები

Select one or more:

- a. toString()
- b. area()
- c. clone()
- d. volume()
- e. print()

პაკეტები

Question 13

რომელი import დირექტივაა სწორი?

Select one or more:

- a. import java.util.Vector;
- b. import java.util.Vector.size();
- c. import java.util.*;
- d. import java.util.Vector.*;

Question 14

რა დაიბეჭდება?

```
package A;
```

```
public class Point {  
    public int x;  
    public int y;  
    public Point(int xx, int yy) {  
        x = xx;  
        y = yy;  
    }  
}
```

```
package A;
```

```
public class Line {  
    Point a = new Point(0, 0);
```

```

Point b = new Point(0, 0);
public Line(Point p1, Point p2) {
a.x = p1.x;
a.y = p1.y;
b.x = p2.x;
b.y = p2.y;
}
public int dist() {
return Math.max(a.x - b.x, a.y - b.y);
}
}

package B;
import A.*;

public class Test {
public static void main(String[] args) {
Point p1 = new Point(1, 0);
Point p2 = new Point(4, 5);
Line l = new Line(p1, p2);
System.out.println(l.dist());
}
}

```

Select one:

- a. 4
- b. -3
- c. 5
- d. 3

Question 15

რომელ სტრიქონში ან სტრიქონებშია შეცდომა:

1. package A;
2. public class Point {
3. private int x;
4. public int y;
5. protected int z;
6. public Point(int xx, int yy, int zz) {
7. x = xx;
8. y = yy;
9. z = zz;
10. }
11. public int vecLength() {
12. return x * x + y * y + z * z;
13. }
14. }
15. package B;
16. import A.*;
17. public class Test {
18. public static void main(String[] args) {

```
19. Point p1 = new Point(3, 4, 1);
20. p1.x = 5;
21. p1.z = 2;
22. System.out.println(p1.vecLength());
23. }
24. }
```

Select one or more:

- a. 20
- b. 21
- c. 7
- d. 19

ინტერფეისები

Question 16

რომელ სტრიქონში/სტრიქონებშია შეცდომა?

```
1. public interface Knopka {
2. void dachera();
3. }
4. public class Bomb implements Knopka {
5. public void dachera() {
6. System.out.println("ar mushaobs");
7. }
8. void boom() {
9. System.out.println("afetqdaaaa");
10. }
11. }
12. public class Test {
13. public static void main(String[] args) {
14. Knopka redKnopka = new Bomb();
15. redKnopka.dachera();
16. redKnopka.boom();
17. }
18. }
```

Select one or more:

- a. 15
- b. 2
- c. 14
- d. 8
- e. 16

განსაკუთრებული სიტუაციების მართვა

Question 17

რა დაიბეჭდება?

```
public class Test {  
    public static void main(String[] args) {  
        int c[] = { 1, 3, 7, 11 };  
        int a = 5;  
        int b = 0;  
        try {  
            int d = c[a * b];  
            System.out.println("End of try block");  
        } catch (ArithmeticException e) {  
            System.out.println("Exception 2");  
        } catch (ArrayIndexOutOfBoundsException e) {  
            System.out.println("Exception 3");  
        } catch (Exception e) {  
            System.out.println("Exception 1");  
        }  
    }  
}
```

Select one:

- a. Exception 1
- b. 15
- c. End of try block
- d. Exception 3
- e. Exception 2

String კლასი

Question 18

სტრიქონი შექმნილია ასე: `String s = new String("xyzyz");` მონიშნეთ, რომელი მეთოდის შესრულების შედეგად შეიცვლება `s` სტრიქონთან დაკავშირებული ობიექტის მნიშვნელობა?

Select one:

- a. `s.replace('z', 'a');`
- b. `s.trim();`
- c. `s.substring(3);`
- d. სტრიქონის მნიშვნელობა არასოდეს არ იცვლება
- e. `s.concat(s);`
- f. `s.append("aaa");`

Question 19

მოცემულია პროგრამის ფრაგმენტი:

1. `String s1 = "abc" + "def";`
2. `String s2 = new String(s1);`

```
3. if (s1 == s2)
4. System.out.println("== succeeded");
5. if (s1.equals(s2))
6. System.out.println(".equals() succeeded");
```

რომელი გამონათქვამია ჭეშმარიტი?

Select one:

- a. მე-4 შესრულდება, ხოლო მე-6 არა
- b. მე-6 შესრულდება, ხოლო მე-4 არა
- c. მე-4 და მე-6 სტრიქონები არ შესრულდება
- d. მე-4 და მე-6 სტრიქონები ორივე შესრულდება