

# ობიექტზე ორიენტირებული დაპროგრამება 2 (Java -ს ბაზაზე)

## II სემესტრი 2017-2018 წელი

### (I შუალედური ტესტირების ნიმუში)

#### კლასების იერარქია

##### კითხვა 1

მოცემულია კლასების იერარქია

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

რომელი გამონათქვამებია ჭეშმარიტი?

აირჩიეთ ერთი:

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

#### ცვლადები მეკვიდრეობითობის დროს

##### კითხვა 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?

აირჩიეთ ერთი:

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

#### მეთოდების გადატვირთვა/გადაფარვა მეკვიდრეობითობის დროს

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

## super-ის გამოყენება კონსტრუქტორში

### კითხვა 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();
}
}

```

აირჩიეთ ერთი:

- a. Base კლასის კონსტრუქტორი SubClass კლასის კონსტრუქტორი
- b. Base კლასის კონსტრუქტორი
- c. SubClass კლასის კონსტრუქტორი



d. არაფერი არ დაიბეჭდება

## super-ის გამოყენება მშობელი კლასის ცვლადზე წვდომისათვის

### კითხვა 5

ჯერ პასუხგაუცემელი

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());
}
}
```

აირჩიეთ ერთი:



a. 12 17



b. 12 12



c. 17 17



d. 17 12

## კლასების ტიპების დაყვანა

### კითხვა 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?

აირჩიეთ ერთი:



a. 10 სტრიქონში შეცდომაა, ვინაიდან ქვეკლასის ტიპის ცვლადს ენიჭება სუპერკლასის ტიპის ობიექტი



b. 9 სტრიქონში შეცდომაა, ვინაიდან სუპერკლასის ტიპის ცვლადს ენიჭება ქვეკლასის ტიპის ობიექტი



c. პროგრამაში შეცდომა არაა



d. Java მკაცრი ტიპიზაციის ენაა ამიტომ სხვადასხვა ტიპის ცვლადების მინიჭება დაუშვებელია

## დინამიკური დისპეტჩინგია

### კითხვა 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?

აირჩიეთ ერთი:



a. 1 2 2 2



b. 2 2 2 1



c. 1 1 2 2



d. 1 2 3 3

## აბსტრაქტული მეთოდები

### კითხვა 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?

აირჩიეთ ერთი:

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

## აბსტრაქტული მეთოდების რეალიზაცია

### კითხვა 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?

აირჩიეთ ერთი:

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

## final მოდიფიკატორი გადაფარვის აკრძალვისათვის

### კითხვა 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?

აირჩიეთ ერთი:

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

## final მოდიფიკატორი მემკვიდრეობითობის აკრძალვისათვის

### კითხვა 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?

აირჩიეთ ერთი:

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

## Object კლასის მეთოდები

### კითხვა 12

მონიშნეთ Object კლასის მეთოდები

Select one or more:

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

## პაკეტები

### კითხვა 13

რომელი import დირექტივაა სწორი?

Select one or more:

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

### კითხვა 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());  
    }  
}
```

აირჩიეთ ერთი:

- a. -3
- b. 9
- c. 4



d. 2

## წვდომის კონტროლი პაკეტებში

### კითხვა 15

რომელ სტრიქონში/სტრიქონებშია შეცდომა?

1. package A;
2. public class Point {
3. private int x;
4. int y;
5. public 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.z = 3;
21. p1.x = 5;
22. p1.y = 2;
23. System.out.println(p1.vecLength());
24. }
25. }

Select one or more:

- a. 7
- b. 20
- c. 21
- d. 22
- e. 19