



# Paradigme de programare

## Model examen

Subiect unic

Nume și prenume: \_\_\_\_\_

Grupa: \_\_\_\_\_

1. Care din următoarele caracteristici nu este valabilă pentru Java?

- a) Orientat pe obiecte
- b) Folosește pointeri
- c) Portabil
- d) Dinamic și extensibil

2. Ce rezultat se va obține prin rularea codului de mai jos?

```
class VariableScope{
    public static void main(String args[])
    {
        int x;
        x = 5;
        {
            int y = 6;
            System.out.print(x + " " + y);
        }
        System.out.println(x + " " + y);
    }
}
```

- a) Eroare la compilare
- b) Eroare la rulare
- c) 5 6 5 6
- d) 5 6 5

3. Ce rezultat se va obține prin rularea codului de mai jos?

```
class Output{
    public static void main(String args[]){
        int arr[] = {1, 2, 3, 4, 5};
        for ( int i = 0; i < arr.length - 2; ++i)
            System.out.print(arr[i] + " ");
    }
}
```

- a) 1 2 3 4 5
- b) 1 2 3 4
- c) 1 2
- d) 1 2 3

4. Ce rezultat se va obține prin rularea codului de mai jos?

```
public class Output{
    public static void main(String args[]){
        String c = "Hello i love java";
        boolean var;
        var = c.startsWith("hello");
        System.out.println(var);
    }
}
```



- }
- a) 0
  - b) true
  - c) 1
  - d) false

5. Ce rezultat se va obține prin rularea codului de mai jos?

```
public class ArrayLists{  
    public static void main(String args[]) {  
        List<String> obj = new ArrayList<>();  
        obj.add("A");  
        obj.add("B");  
        obj.add("C");  
        obj.add(1, "D");  
        System.out.println(obj);  
    }  
}
```

- a) [A, D, C]
- b) [A, B, C]
- c) [A, B, C, D]
- d) [A, D, B, C]

6. Ce rezultat se va obține prin rularea codului de mai jos?

```
final class A{  
    int i;  
}  
class B extends A {  
    int j;  
    void display(){  
        System.out.println(j + " " + i);  
    }  
}  
public class Inheritance {  
    public static void main(String args[]) {  
        B obj = new B();  
        obj.display();  
    }  
}
```

- a) 2 2
- b) 3 3
- c) Runtime Error
- d) Compilation Error

7. Ce rezultat se va obține prin rularea codului de mai jos?

```
class San {  
    public void m1 (int i, float f) {  
        System.out.println(" int float method");  
    }  
  
    public void m1(float f, int i) {  
        System.out.println("float int method");  
    }  
}
```



```
public static void main(String[] args) {  
    San s = new San();  
    s.m1(20, 20);  
}  
}
```

- a) int float method
- b) float int method
- c) eroare la compilare
- d) eroare la rulare

8. Ce rezultat se va obține prin rularea codului de mai jos?

```
class StringTest {  
    public static void main(String args[]) {  
        String obj = "hello";  
        String obj1 = "world";  
        String obj2 = "hello";  
        System.out.println(obj.equals(obj1) + " " + obj.equals(obj2));  
    }  
}
```

- a) false false
- b) true true
- c) true false
- d) false true

9. Când un vector (array) este trecut ca argument unei metode ce primește metoda?

- a) o referință la vector
- b) o copie a vectorului
- c) lungimea vectorului
- d) o copie a primului element

10. Ce rezultat se va obține prin rularea codului de mai jos?

```
class Main {  
    public static void main(String args[]) {  
        try {  
            throw 10;  
        } catch(int e) {  
            System.out.println("Got the Exception " + e);  
        }  
    }  
}
```

- a) Got the Exception 10
- b) Got the Exception 0
- c) eroare de compilare
- d) eroare la rulare

11. Care dintre interfețele de mai jos nu suportă elemente duplicate?

- a) Set
- b) List



- c) Map
- d) nici una din cele de mai sus

12. Ce rezultat se va obține prin rularea codului de mai jos?

```
class UtilitiesTest {  
    public static void main(String[] args) {  
        List<int> intList = new ArrayList<> ();  
        intList.add(10);  
        intList.add(20);  
        System.out.println("The list is: " + intList);  
    }  
}
```

- a) Tipărește: The list is: [10, 20]
- b) Tipărește: The list is: [20, 10]
- c) Generează eroare de compilare
- d) Generează eroare la rulare

13. Ce rezultat se va obține prin rularea codului de mai jos?

```
public static void main(String[] args) {  
    TreeSet<String> ss = new TreeSet<>();  
    ss.add("a");  
    ss.add("b");  
    ss.add("c");  
    ss.add("d");  
    ss.add("e");  
    ss.add("f");  
    ss.add("g");  
    SortedSet<String> ss2 = ss.subSet(ss.first(), ss.floor("e"));  
    Iterator<String> ssitr2 = ss2.iterator();  
    while (ssitr2.hasNext()) {  
        System.out.print(ssitr2.next() + " ");  
    }  
}
```

- a) a b c d e
- b) a b c d
- c) a b c d e f g
- d) Generează eroare la rulare

14. Care este sintaxa corectă pentru referința de metodă conform Java 8?

- a) (args) -> ClassName::methodName
- b) ClassName::methodName(args)
- c) (args) -> methodName::ClassName
- d) methodName::ClassName(args)

15. Ce rezultat se va obține prin rularea codului de mai jos?

```
String string = "22.05.2023";  
DateTimeFormatter formatter =  
    DateTimeFormatter.ofPattern("dd.MM.yyyy");  
LocalDate date = LocalDate.parse(string, formatter);  
System.out.println(date);
```



---

16. Ce metodă poate fi folosită pentru a testa valoarea nulă pe o variabilă de tip Optional?

- a) get()
- b) isPresent()
- c) ofNullable()
- d) map()
- e) toate de mai sus

17. Ce rezultat se va obține prin rularea codului de mai jos?

```
Stream<String> st = Stream.iterate("", (str) -> str + "x");  
System.out.println(st.limit(3).map(str -> str +  
"y").findFirst().orElse(""));
```

- a) xyx
- b) y
- c) xy
- d) xxy

18. Scrieți valorile afișate prin rularea codului de mai jos:

```
public class StreamTest {  
    public static void main(String args[]) {  
        List<Integer> myList = Arrays.asList(10,15,8,49,25,98,32);  
        myList.stream()  
            .filter(n -> n % 2 == 0)  
            .forEach(s -> System.out.print(S + " "));  
    }  
}
```

---

19. Ce rezultat se va obține prin rularea codului de mai jos?

```
public static void main(String[] args) throws InterruptedException,  
ExecutionException {  
    String original = "Test";  
    CompletableFuture<String> cf =  
        CompletableFuture.supplyAsync(() -> original)  
            .thenApply(s -> s.toUpperCase())  
            .thenCompose(upper -> CompletableFuture.supplyAsync(() -> original)  
                .thenApply(s -> s.toLowerCase())  
                .thenApply(s -> upper + " " + s));  
    System.out.println(cf.get());  
}
```

---

20. Ce rezultat se va obține prin rularea codului de mai jos?

```
class Point{  
    int x, y;  
    Point(int x, int y){  
        this.x = x;
```



```
        this.y = y;
    }

    @Override
    public String toString() {
        return "(" + x + ", " + y + ")";
    }
}

public class RxJavaObservableJust {
    public static void main(String[] args) {
        Observable<Point> points = Observable.just(new Point(1,2),
new Point(3,4));
        points.subscribe(p -> {
            System.out.print(p);
        });
    }
}
```

---

---

21. Care din următoarele metode HTTP este folosită pentru a aduce reprezentarea resursei folosind servicii web REST?

- a) GET
- b) DELETE
- c) PUT
- d) OPTIONS

22. Care din următoarele adnotări JAX\_RS leagă parametrii trecuți metodei de un header HTTP?

- a) @PathParam
- b) @HeaderParam
- c) @QueryParam
- d) @MatrixParam

23. Care este codul de stare HTTP adecvat pentru o resursă care a fost ștearsă cu succes într-un API RESTful?

- a) 200 OK
- b) 201 Created
- c) 404 Not Found
- d) 204 No Content

24. Ce este ArC în Quarkus?

---

---

---

25. Ce este Panache în Quarkus?

---

---

---



## Răspunsuri

1.	<b>b</b>
2.	<b>a</b>
3.	<b>d</b>
4.	<b>d</b>
5.	<b>d</b>
6.	<b>d</b>
7.	<b>c</b>
8.	<b>d</b>
9.	<b>a</b>
10.	<b>c</b>
11.	<b>a</b>
12.	<b>c</b>
13.	<b>b</b>
14.	<b>b</b>
15.	<b>2023-05-22</b>
16.	<b>b</b>
17.	<b>b</b>
18.	<b>10 8 98 32</b>
19.	<b>TEST test</b>
20.	<b>(1, 2)(3, 4)</b>
21.	<b>a</b>
22.	<b>b</b>
23.	<b>d</b>
24.	<b>Quarkus ArC este un framework destinat injectiei de dependențe bazat pe specificația Contexts and Dependency Injection (CDI). Este partea a nucleului Quarkus.</b>
25.	<b>Panache este o bibliotecă specifică Quarkus, care ușurează lucrul cu entități persistente bazate pe Hibernate ORM. Entitățile Panache încapsulează atât datele cât și comportamentul.</b>