

Programmazione Avanzata - Test 1 - a.a. 22/23

Test di Programmazione Avanzata, fino alle eccezioni incluse

Punti: 6/10

✓ **Esatto** 1/1 Punti

1. -----

Supponiamo di avere una classe Punto così definita:

```
public class Punto {  
    double x = 1.0;  
    double y = 1.0;  
    Punto(double x, double y) {  
        this.x = x;  
        this.y = y;  
    }  
  
    public double distanza(){  
        return Math.sqrt(x*x+y*y);  
    }  
}
```

e di avere il seguente frammento di codice:

```
Punto p1;  
double r = p1.distanza();
```

☐ r vale 1.0

☐ r vale 0.0☒ viene generato un errore ✓✓ **Esatto** 1/1 Punti

2. -----

Supponiamo di avere il seguente frammento di codice:

```
class A {  
    static void scambia(B uno, B due) {  
        B tmp = uno;  
        uno = due;  
        due = tmp;  
    }  
}  
  
class B {  
    int v;  
  
    B(int v) {  
        this.v = v;  
    }  
  
    public static void main(String[] args) {  
        B b1 = new B(10);  
        B b2 = new B(20);  
        A.scambia(b1, b2);  
        System.out.println(b1.v);  
    }  
}
```

☒ Stampa 10 ✓☐ Viene generato un errore☐ Stampa 20✗ **Non corretto** 0/1 Punti

3. -----

Consideriamo il seguente frammento di codice:

```
class X {  
  
    private int v;  
  
    X(int v) {  
        this.v = v;  
    }  
  
    void m(X altro) {  
        System.out.println(altero.v);  
    }  
  
    public static void main(String[] args) {  
        X x1 = new X(1);  
        X x2 = new X(2);  
        x1.m(x2);  
    }  
}
```

☒ Genera un errore perché v è private

☐ Stampa 2 ✓

✓ **Esatto** 1/1 Punti

4. -----

Il seguente codice

```
class Q {  
    int q;  
  
    Q(int q) {  
        this.q = q;  
    }  
  
    public String toString() {  
        return "q: " + q;  
    }  
  
    public static void main(String[] args) {  
        Q q1 = new Q(1);  
        Q q2 = new Q(2);  
        System.out.println(q1+q2);  
    }  
}
```

- ☒ Genera un errore in fase di compilazione ✓
- ☐ Genera un errore in fase di esecuzione
- ☐ Stampa q: 1q: 2

✗ **Non corretto** 0/1 Punti

5. -----

Dati i due seguenti file

.....

[Due.java](#)

.....


```
public class Due extends Uno {  
    void m1() {  
        super.m1();  
    }  
  
    public static void main(String[]args) {  
        Due d = new Due();  
        d.m1();  
    }  
}
```

.....

[Uno.java](#)

.....

```
public class Uno {  
  
    void m1() {  
        m2();  
    }  
  
    private void m2() {  
        System.out.println("BBB");  
    }  
}
```

- ☐ Genera un errore perché la classe Due tenta di accedere al metodo m1() della classe Uno
- ☐ Stampa BBB 
- ☒ Genera un errore perché m2() è private

✗ **Non corretto** 0/1 Punti

6. -----

Dati i seguenti due file

.....

[Interf.java](#)

.....

```
public interface Interf {  
    void i1();  
}
```

.....

[MiaClasse.java](#)

.....

```
public class MiaClasse implements Interf {  
    public void i1() {  
        // codice  
    }  
    public void m1() {  
        // codice  
    }  
}
```

Indicare quali istruzioni generano un errore:

```
Interf x = new MiaClasse(); // uno  
x.i1(); // due  
x.m1(); // tre  
MiaClasse y = (MiaClasse) x; // quattro  
y.i1(); // cinque  
y.m1(); // sei
```

☐ uno

☒ tre ✓

☐ due

☐ sei

☐ cinque

☐ quattro

✓ **Esatto** 1/1 Punti

7. -----

Data la classe Veicolo così definita

```
public class Veicolo {  
    private String targa;  
    public Veicolo(String t) {  
        targa = t;  
    }  
  
    public String toString(){  
        return targa;  
    }  
}
```

Dire cosa fa il seguente frammento di codice

```
Veicolo v = new Veicolo("AB123");  
Object o = v;  
System.out.println(o);
```

- ☐ Genera un errore
- ☒ Stampa AB123 ✓
- ☐ Stampa Veicolo@hashcode

✗ **Non corretto** 0/1 Punti

8. -----

Si consideri il seguente frammento di codice

```
public class Prova {  
  
    public void met1(double a) {  
        if(a<0) throw new IllegalArgumentException();  
        ...  
    }  
}
```

dopo aver trovato nella documentazione Java le informazioni relative all'eccezione in questione indicare quale delle seguenti affermazioni è vera:

- ☒ non è necessario fare niente ✓

- ☒ è necessario aggiungere al metodo met1() la clausola throws IllegalArgumentException
- ☐ è necessario circondare l'istruzione che lancia IllegalArgumentException con un blocco try-catch

✓ **Esatto** 1/1 Punti

9. -----

Dato il seguente codice

```
import java.util.Scanner;

class MiaEccezione extends Exception {
    MiaEccezione() {
        super();
    }

    MiaEccezione(String a) {
        super(a);
    }
}

public class Ecc {

    public static void main(String[] args) {
        try {
            Scanner s = new Scanner(System.in);
            int i = s.nextInt();
            if(i==1) {
                throw new Exception("i = 1");
            } else if(i==2){
                throw new MiaEccezione("i == 2");
            }
            System.out.println("AAA");
        } catch(MiaEccezione e) {
            System.out.println("BBB");
        } catch(Exception e) {
            System.out.println("CCC");
        }
    }
}
```

indicare quali delle seguenti affermazioni sono vere.

- ☐ Se l'utente inserisce 1 stampa AAACCC
- ☒ Se l'utente inserisce 2 stampa BBB ✓
- ☒ Se l'utente inserisce 1 stampa CCC ✓
- ☒ Se l'utente inserisce 3 stampa AAA ✓
- ☐ Se l'utente inserisce 2 stampa AAABBBCCC

✓ **Esatto** 1/1 Punti

10. Date le seguenti classi, cosa succede se si esegue il comando java B:

```
public abstract class A {  
    public abstract void m1();  
    public void m2() {  
        System.out.println("DDD");  
    }  
    public void m3() {  
        m1();  
    }  
    void m4(){  
        System.out.println("AAA");  
    }  
}
```

```
public class B extends A {  
    public void m1() {  
        System.out.println("BBB");  
        m2();  
    }  
    public void m3() {  
        super.m3();  
    }  
    public void m4() {  
        m3();  
    }  
}
```

```
public static void main(String[] args) {  
    B b = new B();  
    b.m4();  
}
```

- ☐ Entra in un loop infinito
- ☒ Stampa BBBDDD ✓
- ☐ C'è un errore in fase di compilazione, non può essere eseguito
- ☐ Stampa AAABBBDDD

[Torna alla pagina di ringraziamento](#)

Questo contenuto è creato dal proprietario del modulo. I dati inoltrati verranno inviati al proprietario del modulo. Microsoft non è responsabile per la privacy o le procedure di sicurezza dei propri clienti, incluse quelle del proprietario di questo modulo. Non fornire mai la password.

Con tecnologia Microsoft Forms | [Privacy e cookie](#) | [Condizioni per l'utilizzo](#)