

Cognome

Nome

Matricola

1 (8 punti)

Data la classe **Vendemmia**:

<pre> public class Vendemmia { public int carico; public float mosto; public int pigiatura; private static int caricoTotale= 0; private static float mostoTotale; private static int numeroPigiature= 0; /* Costruisce un oggetto per la vendemmia di un agricoltore */ public Vendemmia() { carico = 0; pigiatura = 0; mosto = 0; } /* Costruisce un oggetto per la vendemmia di un agricoltore con carico iniziale*/ public Vendemmia(int caricoUva) { carico = caricoUva; caricoTotale += caricoUva; pigiatura = 0; mosto = 0; } /* metodo per aggiungere un carico di uva da pigiare */ public int carica(int caricoUva) { carico = caricoUva; caricoTotale += carico; return carico; } </pre>	<pre> /* metodo per effettuare la pigiatura. Si tiene conto della perdita del 10% in raspi e il parametro rappresenta il rendimento dell'uva raccolta */ public void pigia(int rendimento) { pigiatura = (int) (carico * 0.9); mosto= pigiatura * rendimento/100; mostoTotale += mosto; numeropigiature++; } /* metodo per produrre una descrizione opportuna della vendemmia di ciascun agricoltore */ public String toString() { return "vendemmia: " + carico + ", mosto: " + mosto; } /* metodo per leggere la quantità totale di uva vinificata */ public static float getMosto() { return mostoTotale; } /* metodo statico per leggere il numero di pigiature fatte*/ public static int getPigiature(){ return numeroPigiature; } /* metodo per leggere la raccolta totale d'uva */ public static int getCarico() { return caricoTotale; } </pre>
---	--

Istanziare un oggetto della classe **Vendemmia** di nome **miaVendemmia** per predisporre la raccolta futura:

Istanziare un oggetto **tuaVendemmia** con un carico di uva di 200 kg:

Raccolti 100 kg di uva caricateli per la pigiatura della mia vendemmia, invocando il metodo:

Procedere con la pigiatura della mia vendemmia, tenendo conto che il rendimento della mia uva e' dell'70%:

Ora e' il turno della pigiatura della tua vendemmia che rende l'80%:

In un altro campo raccolgo altri 50 kg di uva da vinificare che rende il 90%. Quindi invoco i metodi:

Stampo il rapporto della mia e della tua vendemmia ora, ottenendo:

```
System.out.println(miaVendemmia);
System.out.println(tuaVendemmia);
```

Invocare i metodi seguenti e calcolare il valore che riportano:

```
Vendemmia.getPigiature();
Vendemmia.getMosto();
Vendemmia.getCarico();
```

2 (3 punti)

Dato l'oggetto

```
Vendemmia vendemmia2005 = new Vendemmia(300);
```

Dire se sono corrette le istruzioni:

```
int r = vendemmia2005.carica(100);      SI    NO
vendemmia2005.pigia(80);                SI    NO
```

Valutate le seguenti espressioni:

```
r
vendemmia2005.carico;
Vendemmia.getMosto();
```

3 (1 punto)

Dire qual è la caratteristica di Java che consente di definire due costruttori **Vendemmia**:

Dire come si distinguono i due costruttori:

4 (3 punti)

Assumendo la dichiarazione:

```
Random rand = new Random();
```

Indicare il range dei valori delle seguenti dichiarazioni:

```
rand.nextInt() % 9;
(int) (Math.random() * 4);
```

Inoltre scrivere un'istruzione per produrre valori pseudo-casuali nell'intervallo:

```
[-2, 4]   usando l'oggetto rand
[4, 12]   usando il metodo random() della classe Math
```

5 (3 punti)

Esprimere in linguaggio Java la seguente condizione, usando gli operatori di relazione e quelli logici:
il numero n deve essere maggiore di 10 ma non di 30

Esprimere in linguaggio Java la negazione della condizione precedente senza introdurre l'operatore di negazione (applicare la legge di De Morgan).

6 (2 punti)

Data la stringa:

```
String riga = new String("alta, solenne, vestita di nero");
```

calcolare l'output delle istruzioni:

```
riga.length();
riga.substring(4, 9).length();
riga.substring(3, 10).toUpperCase();
riga.replace('e', 'a').substring(3, 14);
String nuova = riga.substring(8, 13).replace('n', 'm');
System.out.println (nuova);
System.out.println (riga);
```

7 (2 punti)

Assumendo le dichiarazioni:

```
int x;
String parola = new String("Ei fu");
```

Indicare l'ordine di valutazione degli operatori nelle seguenti espressioni, mettendo il numero corrispondente sotto al simbolo dell'operatore (considerate anche l'operatore dot).

```
x = 2 * parola.length() + 4

x = parola.length()+ 2 * 3

x = 2 * parola.toUpperCase().length()
```

8 (3 punti)

Date le variabili:

```
int a = 1, b = 5;
```

dopo aver eseguito separatamente i due blocchi di istruzioni:

<pre>a = a + b; b += a + --b;</pre>	<pre>a = b-- * ++a; b += a - b--;</pre>
-------------------------------------	---

calcolare il valore di :

<pre>a: b:</pre>	<pre>a: b:</pre>
------------------	------------------

9 (5 punti)

Data l'espressione booleana:

`numero < 10 || !finito`

compilarne la tabella di verità.

<code>numero < 10</code>	<code>finito</code>	<code>!finito</code>	<code>numero < 10 !finito</code>

Specificare una possibile coppia di valori delle variabili `numero` e `finito` per rendere falsa la condizione:

Se la condizione fosse usata in un ciclo (si entra e si resta nel ciclo se la condizione è vera):

```
Finché  
    <condizione>  
Fai  
    <istruzioni; numero++; finito = !finito>
```

nell'ipotesi che, prima d'iniziare il ciclo, `numero` sia 8 e `finito` sia false, dire quante volte viene eseguito il blocco d'istruzioni :

```
<istruzioni; numero++; finito = !finito>
```