

Programmazione ad Oggetti – Verifica 1

Matricola:.....

1. Dato il seguente frammento di codice:

```
String s;  
System.out.println(s);
```

quale di questi risultato otteniamo:

- Non stampa nulla
- Stampa un a-capo e basta
- Stampa "null"
- Genera un errore di sintassi
- Genera un errore di run-time

2. Dato il seguente frammento di codice:

```
String s1 = new String("A");  
String s2 = new String("A");
```

quale di queste condizioni è true:

- s1 == s2
- s1.equals(s2)
- s1 == "A"
- s2 == "B"
- s1 == null

3. Dato il seguente frammento di codice:

```
String array[] = new String[10];
```

quale di queste espressioni è true

- array[1] == null
- array[1] == ""
- array[1] == new String("")
- array[1].equals("")
- array[1].equals(new String(""))

4. Dato il seguente frammento di codice:

```
String array[] = new String[10];  
array[0] = "A";  
array[2] = "B";  
array[3] = "C";
```

che valore assume array.length ?

- 3
- 4
- 9
- 10
- 11

5. Quale delle seguenti istruzioni di
inizializzazione di un array di interi è sbagliata?

- int[] v = {1,2,3};
- int[] v1 = new int[]{1,2,3};
- int[] v3 = new int[3];
- int[] v2 = new int[3]{1,2,3};

Soluzioni

1. Risposta 4

```
String s;  
System.out.println(s);
```

Il compilatore Java genera un errore di sintassi in corrispondenza della seconda istruzione: la variabile “s” non è stata inizializzata.

Java impone che all’interno di un singolo metodo le variabili locali siano inizializzate esplicitamente prima di essere utilizzate (lette o passate come parametro).

2. Riposta 2

```
String s1 = new String(“A”);  
String s2 = new String(“A”);
```

Vengono istanziati due oggetti distinti, quindi i due reference “s1” ed “s2” puntano a oggetti diversi (ovvero $s1 \neq s2$) ma i due oggetti rappresentano stringhe identiche (“A”) quindi il metodo `equals()` restituisce `true`.

3. Risposta 1.

```
String array[] = new String[10];
```

Quando viene allocato un array di reference (non tipi primitivi) tutti gli elementi vengono automaticamente inizializzati a “null”. Quindi l’array dell’esempio conterrà 10 riferimenti che non puntano a nessun oggetto.

4. Risposta 4.

```
String array[] = new String[10];  
array[0] = “A”;  
array[2] = “B”;  
array[3] = “C”;
```

Indipendentemente dal contenuto degli elementi, l’attributo `length` degli array contiene sempre il numero di elementi allocati tramite l’operatore `new`.

5. Risposta 4.

```
int[] v2 = new int[3]{1,2,3}; // errore doppia definizione della lenght
```

Java permette di inizializzare il contenuto di un array con una lista di inizializzazione, in tale caso il numero di elementi viene dedotto dal numero di elementi della lista. Se viene contemporaneamente specificato tra `[]` il compilatore segnala un errore.