

**[Quiz]** Definire una piccola gerarchia di classi per rappresentare quiz di vario genere (a scelta multipla, a risposta numerica e a risposta testuale) rispettando le seguenti indicazioni.

- a. (*max 9 punti*) Definire una classe base `Quiz` e le sottoclassi `QuizMulti`, `QuizNumber` e `QuizWord`. La classe `Quiz` gestisce i dati comuni: il testo del quiz (una stringa) e il punteggio (un intero). La sottoclasse `QuizMulti` rappresenta quiz a scelta multipla e deve quindi gestire un array di stringhe rappresentanti le possibili scelte e la risposta corretta, un `char` con il seguente significato: 'A' indica la prima scelta, 'B' indica la seconda scelta, e così via. La sottoclasse `QuizNumber` rappresenta quiz con risposta numerica e deve gestire un array di possibili risposte corrette (un array di `int`). La sottoclasse `QuizWord` rappresenta quiz con risposta data da una parola e deve gestire un array di possibili risposte corrette (un array di stringhe). Definire un metodo `getScore` che ritorna il punteggio del quiz. Definire un metodo `isCorrect` che prende in input una stringa `answer` e ritorna `true` o `false` a seconda che `answer` sia una risposta corretta o meno. Nel caso della classe `Quiz` il metodo ritorna sempre `false`, le sottoclassi devono ridefinire il metodo. Per `QuizMulti` solo il primo carattere di `answer` va considerato. Per `QuizNumber` la stringa `answer` va interpretata come un intero (si può usare il metodo `Integer.parseInt()`) e la risposta è corretta solo se l'intero è presente nell'array delle risposte corrette. Per `QuizWord` la risposta è corretta solo se `answer` è uguale a una delle stringhe dell'array delle risposte corrette. Definire inoltre un metodo `print` che prende in input un intero `n` e stampa il quiz come nel seguente esempio (in questo caso `n = 3`, il quiz è a scelta multipla, il testo è "Come si chiama il satellite del pianeta Terra?", il punteggio è 5 e le possibili scelte sono "Giove", "Luna", "Marte", "Urano"):

Quiz 3 (score 5)

Come si chiama il satellite del pianeta Terra?

- A. Giove
- B. Luna
- C. Marte
- D. Urano

- b. (*max 9 punti*) Definire una classe `QuizTest` con un metodo statico `readQuizFile` che prende in input il pathname di un file contenente una batteria di quiz e ritorna in un array di oggetti `Quiz` i quiz letti dal file. Ogni linea del file definisce un quiz secondo il seguente formato: `X#T#S#R1#R2#...#RK`. `X` è un carattere che indica il tipo del quiz: 'M' scelta multipla, 'N' a risposta numerica e 'W' a risposta testuale. `T` è una stringa che definisce il testo del quiz e `S` è una stringa che rappresenta il punteggio. Il significato delle altre stringhe `R1, ..., RK` dipende dal tipo di quiz. Se `X = 'M'` allora `R1` è un singolo carattere che rappresenta la risposta corretta e `R2, ..., RK` sono le scelte multiple. Se `X = 'N'` allora `R1, ..., RK` sono stringhe che rappresentano le risposte numeriche corrette. Se `X = 'W'` allora `R1, ..., RK` sono le risposte corrette.

c. (*max 6 punti*) Definire per la classe `QuizTest` un metodo `main` che, usando il metodo `readQuizFile`, legge dalla linea di comando i pathname di due file: il primo contiene una batteria di quiz secondo il formato specificato al punto (b) e il secondo contiene delle risposte per tali quiz con una risposta (stringa) per linea. Poi il metodo `main` deve stampare per ogni quiz, il testo del quiz, la relativa risposta, l'indicazione se la risposta è corretta o meno. Infine deve stampare il punteggio totale (somma dei punteggi ottenuti in tutti i quiz). Ad esempio se i due file contengono:

FILE DEI QUIZ:

```
M#Come si chiama il satellite della Terra?#A#B#Giove#Luna#Marte
W#Una provincia dell'Umbria.#3#Terni#Perugia
N#Un numero primo tra 30 e 40.#2#31#37
```

FILE DELLE RISPOSTE:

```
C
Terni
31
```

Allora il `main` stampa:

```
Quiz 1 (score 4)
Come si chiama il satellite della Terra?
A. Giove
B. Luna
C. Marte
Answer: C NOT CORRECT
Quiz 2 (score 3)
Una provincia dell'Umbria.
Answer: Terni CORRECT
Quiz 3 (score 2)
Un numero primo tra 30 e 40.
Answer: 31 CORRECT
Total score: 5
```

**[Errori]** (*max 4 punti*) Il seguente codice Java contiene uno o più errori. Trovare gli errori e spiegarli. In particolare, dire per ogni errore se si verifica in compilazione o durante l'esecuzione.

```
class Value {
    private Object value;

    public Value(Object v) { value = v; }

    public void setValue(String v) { value = v; }
    public Object getValue() { return value; }
}

class ValueString extends Value {
    public ValueString(String s) { super(s); }
}

public class Test {
    public static void main(String[] args) {
        Value[] val = new Value[2];
        val[0] = new Value(new Value("A"));
        val[1] = new ValueString("B");
        Value v = val[0].getValue();
        Object obj = val[0].getValue();
        String s = (String)obj;
    }
}
```

```
String ss = (String)val[1].getValue();  
    }  
}
```

**[Insiemi di anagrammi]** Due stringhe  $s_1$  e  $s_2$  sono *anagrammi* l'una dell'altra se ogni carattere di  $s_1$  o di  $s_2$  occorre lo stesso numero di volte in  $s_1$  e in  $s_2$ .

- a. (*max 6 punti*) Definire una classe `Anagrammi` per rappresentare insiemi di anagrammi. Il costruttore prende in input una stringa e costruisce l'insieme composto da tale stringa. Definire un metodo `aggiungi` che prende in input una stringa  $s$  e se  $s$  è un anagramma delle stringhe nell'insieme ed è diversa da tutte le stringhe dell'insieme allora la aggiunge all'insieme e ritorna `true`, altrimenti non fa nulla e ritorna `false`.
- b. (*max 6 punti*) Definire un metodo `seleziona` della classe `Anagrammi` che prende in input un array di stringhe  $sA$  e elimina dall'insieme tutte le stringhe che non appaiono nell'array  $sA$ .