

Programmazione in Java e Gestione della Grafica - A.A. 2017-2018 -

PreAppello - I modulo - febbraio 2018

ISTRUZIONI:

- Possono essere consegnati programmi che realizzano solo parzialmente il testo, tuttavia i programmi consegnati devono essere "compilabili". Non verranno valutati programmi che presentano errori alla compilazione.
- Creare una cartella `<CognomeNome>PreAppello` e mettere al suo interno i files realizzati (soltanto .java)
- Copiare la cartella nella chiavetta USB del docente.

Esercizio 1: Analisi di vettori

Si diano le seguenti definizioni:

- Un vettore di dimensione n ha un *bordo di lunghezza b* se i primi b simboli del vettore sono uguali ordinatamente e specularmente agli ultimi b simboli. (Il bordo é quindi la sequenza che si legge allo stesso modo da sinistra verso destra e da destra verso sinistra). Ad esempio il vettore

0	1	1	1	0	1	0	1	1	0
---	---	---	---	---	---	---	---	---	---

 ha sia un bordo di lunghezza 1 (0) che un bordo di lunghezza 3 (011).
- Un vettore di dimensione n contiene un *segmento di lunghezza s* se esistono s posizioni consecutive che contengono lo stesso simbolo. Ad esempio il vettore

0	1	1	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---	---	---

 contiene un segmento di lunghezza 4 (di simboli 0).

Scrivere una classe Java `Vettore.java` contenente:

- un metodo statico `maxBordo` che, ricevuto come parametro un array di interi a , restituisca un intero corrispondente alla lunghezza del più lungo bordo in a .
- un metodo statico `maxSegmento` che, ricevuto come parametro un array di interi a , restituisca un intero corrispondente alla lunghezza del più lungo segmento in a .
- un metodo `main` che inizializza un vettore di lunghezza 20 con valori random booleani (0 e 1), lo stampa sullo schermo, poi richiama il metodo `maxBordo` e `maxSegmento` e poi stampa la lunghezza del massimo bordo e del massimo segmento del vettore.
NOTA: le istruzioni di stampa devono essere nel `main` e non nei metodi `maxBordo` e `maxSegmento`.

Esercizio 2: Giocare con i vettori

Definire le seguenti classi: (max 15 punti)

- `public class Persona`
Campi: - `nome` tipo: String (il nome della persona)
 - `vettoreX` vettore random di 20 elementi a valori booleani
Metodi: - tutti i metodi standard: costruttore(i), metodi set e get , metodo toString.
- `public class PersonaB extends Persona`
Metodi: - (costruttore)
 - `public int calcolaB()` che calcola la lunghezza del massimo bordo di `vettoreX` (utilizzare il codice del metodo `maxBordo` dell'esercizio precedente - trasformare il metodo statico in metodo di classe)
- `public class PersonaS extends Persona`
Metodi: - (costruttore)
 - `public int calcolaS()` che calcola la lunghezza del massimo segmento di `vettoreX` (utilizzare il codice del metodo `maxSegmento` dell'esercizio precedente - trasformare il metodo statico in metodo di classe)

Scrivere il seguente programma (classe contenente il *main*). (max 15 punti)

- contiene metodo: `public static Persona Sfida (PersonaB b, PersonaS s)` che simula la sfida tra le due persone. Ognuno produce il valore corrispondente al proprio metodo `Calcola` e chi ha un valore maggiore vince la sfida e viene restituito come vincitore.
- contiene metodo: `public static void ProclamaVincitore (Persona g)` che:
 1. scrive sullo schermo se il vincitore é di tipo `PersonaB` o di tipo `PersonaS`
 2. scrive sullo schermo il suo nome
- contiene metodo: `public static void main(String[] args)`che:
 1. inizializza una `PersonaB` e una `PersonaS` e stampa i dati corrispondenti sullo schermo (nome e vettore).
 2. Esegue un turno di `Sfida` (richiamando il metodo corrispondente) tra i due oggetti creati.
 3. Proclama il vincitore (richiamando il metodo corrispondente)
 4. Stampa una coppa usando come parametro di larghezza il valore che ha decretato il vincitore (di `max bordo` o `max segmento`). In caso di pareggio, non verrà stampata la coppa.
- **VARIANTE** (max 5 punti)
Si effettua una gara consistente di 10 turni: ad ogni turno viene inizializzato un nuovo vettore random per ogni giocatore e viene effettuata la sfida. Il vincitore sarà colui che vince più sfide.