

Esercitazione 4

Lettura file e classi

AA2016/17

File

- ▶ Nel package *prog.io* esistono le seguenti classi per manipolare i file:
 - **FileInputManager(String nomefile)**: lettura del file
 - **FileOutputManager(String nomefile)**: scrittura del file
- ▶ FileInputManager mette a disposizione i seguenti metodi:
 - `public static boolean exists(String nomefile)`: true se esiste il file
 - `public String readLine()`: legge una riga di testo dal file

Esercizio 1

- ▶ Il programma deve leggere un file contenente un testo. Contare quante vocali sono contenute all'interno del testo e scrivere a video il testo in maiuscolo.
- !!! Il file di testo è da salvare nella cartella del progetto

Esercizio 2

- ▶ Leggi un testo da un File (usa file input manager)
- ▶ Memorizza le parole nel testo in una sequenza ordinata
 - ▶ Usa `readLine`, `split`, `SequenzaOrdinata`
- ▶ Ristampa le parole in ordine

- ▶ Nota: mette il file di testo nella root del progetto e riferisciti ad esso semplicemente con il suo nome (con anche estensione)

Esercizio 3

- ▶ Modifica l'esercizio 3B dell'esercitazione del giorno 05/04/2017
- ▶ Leggi i dati delle figure da un file di testo (una figura per riga)
 - ▶ Leggi le righe
 - ▶ Per ogni riga aggiungi l'elemento alla sequenza/array di figure
 - ▶ Trova la figura con area massima e perimetro massimo
- ▶ Esempio contenuto file:
 - R,5,7
 - Q,8
 - C,5

Classi e oggetti

- ▶ Classe è una collezione di uno o più oggetti contenenti un insieme uniforme di attributi e servizi, insieme ad una descrizione circa come creare nuovi elementi della classe stessa (Edward Yourdan);
- ▶ Un oggetto è dotato di stato, behavior ed identità; la struttura ed il comportamento di oggetti simili sono definiti nelle loro classi comuni; i termini istanza ed oggetto sono intercambiabili (Grady Booch).

Esercizio 4

- ▶ Creare una classe ContoCorrente in cui si crea un conto e si implementano le seguenti operazioni:
 - Preleva
 - Ricarica
 - CalcolaInteressi (0,0001%)

Ogni conto è costituito da:

- Numero di conto
 - Nome proprietario
 - Importo iniziale
-
- Fare una classe di prova contenente il metodo main per testare i metodi della classe ContoCorrente

Esercizio 5

- ▶ Definisci una classe Parallelogramma (P) con i seguenti attributi: diagonale maggiore (D) diagonale minore (d) e angolo (p) che la diagonale maggiore forma rispetto l'orizzontale.
Deve essere possibile calcolare, i lati, l'area e perimetro del parallelogramma.
- ▶ Scrivere un metodo Main per testare il programma con alcuni parallelogrammi d'esempio

Campi e metodi statici

- ▶ Campi statici: campi associati all'intera classe e non ai singoli oggetti.
- ▶ Metodi statici: metodi che manipolano esclusivamente un'informazione comune a tutta la classe e sono indipendenti dagli oggetti della classe stessa. Il metodo potrebbe essere invocato anche prima di costruire istanze della classe.

Esercizio 6

- ▶ Un trapano verticale si può muovere lungo le direzioni x,y di un piano per poi fermarsi ed eseguire un foro.
- ▶ L'operazione di foratura e di traslazione lungo gli assi xy è subordinata al fatto che il motore del trapano sia acceso o spento.
- ▶ Inizialmente il trapano è in posizione (0,0). La massima posizione è (1000,1000) (usa variabili static).
- ▶ Realizza una classe Trapano e una classe 'Motore' che gestisca lo spegnimento e l'accensione del motore del trapano. Ogni trapano ha un motore.
- ▶ Implementa i seguenti metodi nella classe appropriata
- ▶ `public void fora()`: accende il motore se non è acceso e manda un messaggio dopo 2 secondi che il foro è terminato
 - ▶ Usa `Thread.sleep()`
- ▶ `public void muovi(int dx,int dy)`: sposta il trapano di dx e dy se è possibile (controlla i limiti) e stampa la nuova posizione
 - ▶ Se non è possibile visualizza un messaggio
- ▶ Accendi e spegli il motore
- ▶ Fai anche un main in cui provi i diversi metodi

Esercizio 7

- ▶ Si scriva un programma che registri i risultati delle partite di una squadra di calcio.
- ▶ Creare una classe Partita in cui si deve indicare
 - il nome della squadra e il numero di partite fatte in un campionato (sono due campi statici).
 - la squadra avversaria e il punteggio delle squadre nella partita
 - Aggiungere un metodo che determina chi ha vinto dati i punteggi delle squadre

Fare un metodo main in cui si acquisisce una serie di partite (da leggere da un file di testo). Al termine dell'inserimento si vogliono visualizzare le partite in cui la squadra ha vinto, quelle in cui ha pareggiato e quelle in cui ha perso utilizzando il metodo dichiarato nella classe Partita (nome avversario, punteggio, risultato: «Ha vinto», «Ha perso», «Ha pareggiato»).