

Tema 3 POO

Competiții Sportive (Design Patterns)

Responsabil: Dragoș Corlătescu

Introducere

Anul acesta faceți parte din managementul unei gale de sport în cadrul căreia să vor desfășura mai multe competiții sportive precum fotbal, baschet și handbal. Responsabilitatea voastră este de a implementa un sistem care asigură atât înscrierea echipelor, cât și gestionarea meciurilor și a rezultatelor.

Cerințe

Sistemul/tema are următoarele cerințe care trebuie implementate având în minte conceptele legate de POO deja învățate, dar și folosind design pattern-urile indicate:

Înscrierea echipelor (25p)

Sistemul vostru trebuie să permită înscrierea echipelor. Pentru aceasta veți implementa o ierarhie de clase începând de la clasa **Team**, care va fi extinsă de alte 3 subclase:

FootballTeam, BasketballTeam, HandballTeam.

Clasa părinte va conține câmpurile:

- *teamName*, reprezentând numele echipei
- *gender*, reprezentând tipul de echipă: masculină/feminină
- *numberOfPlayers*, reprezentând numărul de jucători dintr-o echipă
- *players*, reprezentând lista jucătorilor din echipa respectivă.

Jucătorii vor trebui reprezentați într-o clasă **Player**, care va conține câmpurile *name*, numele jucătorului și respectiv *score*, reprezentând valoarea acestuia.

I/O: Pentru testarea acestei cerințe programul vostru va primi în linie de comandă 4 parametri:

inscriere *nume_fisier_intrare !random_string! nume_fisier_iesire*

“**inscriere**” este un cuvânt fix care determină cerința.

Fișierul de intrare va conține mai multe echipe de forma:

type, teamName, gender, numberOfPlayers

namePlayer1, scorePlayer1

...

type poate fi: **football, basketball** sau **handball**

Acel *!random_string!* este un cuvânt care poate fi orice (ex: "POO") pe care-l veți ignora.

Ușurează implementarea makefile-ului, ambele cerințe având în acest caz același număr de parametri.

Fișierul de ieșire va avea formatul următor pentru fiecare echipă:

{teamName: *teamName*, gender: *gender*, numberOfPlayers: *numberOfPlayers*, players: [{name: *namePlayer1*, score: *scorePlayer1*}, ...]}

...

Exemplu

inscriere *echipe1.in inscriere_echipe.out*

<i>echipe1.in</i>	<i>inscriere_echipe.out</i>
football, EchipaRacheta, masculin, 11 Elon1, 23 Elon2, 11 Elon3, 25 Elon4, 123 Elon5, 34 Elon6, 12 Elon7, 42 Elon8, 9 Elon9, 8 Elon10, 99 Elon11, 1 basketball, The F Lakers, feminin, 5 Lebrona1, 100 Lebrona2, 100 Lebrona3, 100 Lebrona4, 100 Lebrona5, 100	{teamName: EchipaRacheta, gender: masculin, numberOfPlayers: 11, players: [{name: Elon1, score: 23}, {name: Elon2, score: 11}, {name: Elon3, score: 25}, {name: Elon4, score: 123}, {name: Elon5, score: 34}, {name: Elon6, score: 12}, {name: Elon7, score: 42}, {name: Elon8, score: 9}, {name: Elon9, score: 8}, {name: Elon10, score: 99}, {name: Elon11, score: 1}]} //// new line //// {teamName: The F Lakers, gender: feminin, numberOfPlayers: 5, players: [{name: Lebrona1, score: 100}, {name: Lebrona2, score: 100}, {name: Lebrona3, score: 100}, {name: Lebrona4, score: 100}, {name: Lebrona5, score: 100}]}

Important ! Design pattern-urile care trebuie folosite în implementarea acestei cerințe cel puțin o dată sunt: **Singleton și Factory**.

Competiții (65p)

Pentru partea de competiție va trebui să implementați o clasă care să o reprezinte. Ea va conține câmpurile:

- *type*, tipul competiție sau ce fel de sport se joacă
- *gender*, cui îi este dedicată masculin/feminin
- *teams*, o listă cu toate echipele care participă la acea competiție
- *clasament*

După ce toate echipele au fost adăugate, competiția poate începe. Pentru determinarea clasamentului final fiecare echipă va juca cu toate celelalte echipe o singură dată. Câștigătorul unui meci va fi desemnat în funcție de scorul fiecărei echipe, care va fi accesat cu ajutorul design pattern-ului **Visitor**.

Cazurile posibile sunt următoarele pentru calcularea scorului unei echipe:

- fotbal, masculin = scorul celui mai bun jucător x 2 + suma scorurilor celorlalți jucători
- fotbal, feminin = scorul celui mai slab jucător x 2 + suma scorurilor celorlalți jucători
- baschet, indiferent de gen = media scorurilor jucătorilor
- handbal, masculin = suma scorurilor jucătorilor
- handbal, feminin = produsul scorurilor jucătorilor

În urma unui meci, echipa câștigătoare va primi 3 puncte, iar echipa pierzătoare va primi 0 puncte. În caz de egalitate, ambele echipe vor primi 1 punct. Cu aceste puncte se va actualiza și clasamentul, iar toate echipele vor fi “notificate” de această actualizare prin intermediul design pattern-ului **Observer**.

La sfârșit, se vor afișa primele 3 echipe (din competiție) și fiecare echipă își va afișa locul ocupat.

I/O: Pentru testarea acestei cerințe programul vostru va primi în linie de comandă 4 parametri:

competitie *nume_fisier_intrare_echipe* *nume_fisier_intrare_competitie*
nume_fisier_iesire

“**competitie**” este un cuvânt fix care determină cerința.

Primul fișier de intrare cu echipe va avea formatul fișierului de intrare de la cerința anterioară.

Al doilea fișier de intrare va defini competiția și va avea următorul format:

type, gender
team name 1
team name 2

...

Se garantează că echipele prezente în acest fișier există în primul fișier. **NU** se garantează însă că tipul și/sau genul echipelor este cel conform cu al competiției. În acest caz, echipa nu trebuie adăugată în competiție.

Fișierul de ieșire va conține pe primele 3 linii echipele care la finalul competiției au ieșit pe locurile 1, 2 și 3. Pe liniile următoare fiecare echipă în ordinea introducerii lor în competiție își vor afișa poziția ocupată în felul următor:

Echipa *team name 1* a ocupat locul *locul_conform_clasamentului*

Echipa *team name 2* a ocupat locul *locul_conform_clasamentului*

...

Exemplu

competitie echipe.in competitie.in rezultate.out

<i>echipe.in</i>	<i>competitie.in</i>
basketball, theTeam1, masculin, 5 P11, 10 P12, 10 P13, 10 P14, 10 P15, 10 basketball, theTeam2, masculin, 5 P21, 12 P22, 12 P23, 12 P24, 12 P25, 12 basketball, theTeam3, masculin, 5 P31, 9 P32, 9 P33, 11 P34, 9 P35, 11 football, outsideTeam, feminin, 1 Maradona, 1000	basketball, masculin theTeam3 theTeam2 outsideTeam (comentariu: nu va fi adaugata) theTeam1

rezultate.out
theTeam2 theTeam1 theTeam3 Echipa <i>theTeam3</i> a ocupat locul 3 Echipa <i>theTeam2</i> a ocupat locul 1 Echipa <i>theTeam1</i> a ocupat locul 2

Bonus (20p)

Pentru bonus va trebui să implementați calculul scorului unei echipe folosind design pattern-ul **Strategy**, păstrând, bineînțeles și implementarea de **Visitor** precedentă.

Trimitere și notare

- Punctaj:
 - Înscriere (prima cerință): 25 puncte
 - Competiție (a doua cerință): 65 puncte
 - Bonus: 20 puncte
 - Coding style, Readme și javadoc: 10 puncte
 - **Total: 120 puncte**
- Tema va trimite pe vmchecker într-o arhivă care va conține fișierul Makefile, sursele, javadoc-ul și fișierul readme. Makefile-ul trebuie să conțină reguli de *build*, *run* și *clean*.
 - Makefile-ul trebuie să poată primi parametri în linie de comandă.
 - **COMANDA**=numele_comenzii(inscriere/competitie)
 - **IN1**=primul_fisier_de_intrare
 - **IN2**=al_doilea_fisier_de_intrare(sau_un_cuvant_random_pentru_prima_cerinta)
 - **OUT**=fișierul_de_output
 - <https://stackoverflow.com/questions/2826029/passing-additional-variables-from-command-line-to-make>
- Jumătate din teste vor fi publice, iar jumătate vor fi private.

Informatii importante!

- Este **obligatorie** folosirea design pattern-urilor prezentate în enunț pentru rezolvarea temei. Nerespectarea acestei reguli poate duce la depunctări de până la 100%.
- Nu este obligatorie folosirea numelor câmpurilor/claselor prezentate în enunț, dar este obligatorie implementarea ierarhiilor specificate.

Succes!