



PROGRAMACIÓ 1

CURS: 2019/2020-Q2

ACTIVITAT 1

1 Disseny modular en C++: Gestió d'una competició de robots

Els organitzadors d'una competició robòtica dissenyen un conjunt de missions que un robot ha de dur a terme dins una àrea de joc en un temps limitat. Un exemple de missió és que el robot pugi sobre un pont o que aconsegueixi desplaçar una peça dins l'àrea de joc. Cada missió, depenent de la seva dificultat, dona una quantitat de punts sempre que el robot aconsegueixi completar-la amb èxit.

A una competició s'hi inscriuen una sèrie d'equips. Cada equip dissenya i programa el seu robot perquè realitzi les missions proposades o, almenys, les que estratègicament considera que pot abordar. L'equip guanyador de la competició és aquell que obté més punts¹.

La [primera tasca](#) d'aquesta activitat serà especificar les classes `Missio`, `Equip` i `Competicio` que permetin representar els components d'una competició de robots com es descriuen a continuació.

Una `Missio` s'identifica amb un nom i consta d'una descripció i una puntuació. Exemple de `Missio`:

```
missio9
factor_de_seguretat
40
```

on `missio9` és el nom de la missió, `factor_de_seguretat` és la seva descripció i `40` és la quantitat de punts que es pot obtenir amb aquesta missió si se supera amb èxit.

Un `Equip` es representa amb un nom, que l'identifica, i pot tenir o no puntuació. En el cas de tenir puntuació, l'equip conté la puntuació final que ha obtingut en la competició i la missió amb què ha obtingut més punts. Exemple d'`Equip` amb puntuació:

```
equip6
220
missio9
```

¹Diffícilment dos equips aconsegueixen la mateixa puntuació però, en cas d'empat, suposarem que guanya el primer en ordre d'inscripció.



on `equip6` és el nom de l'equip, 220 és la puntuació final de l'equip en la competició i `missio9` el nom de la missió que li ha donat més punts (entre totes les que ha realitzat el seu robot).

Exemple d'Equip sense puntuació:

```
equip3  
0
```

on `equip3` és el nom de l'equip i 0 indica que no té puntuació.

Una **Competició** s'identifica amb la data i el lloc de celebració, i està formada per un conjunt d'equips participants. Exemple de **Competició**:

```
08/02/2020  
EEBE-UPC  
equip1  
equip2  
equip3  
equip4
```

on 08/02/2020 i EEBE-UPC són la data i el lloc de celebració de la competició a la que s'hi han inscrit 4 equips `equip1...equip4`.

Per cada classe, haureu de definir:

- almenys dos constructors (per exemple, un constructor pot crear un objecte de la classe buit i un altre constructor pot crear un objecte de la classe amb els valors d'alguns dels seus atributs rebuts per paràmetre)
- un destructor
- modificadors dels atributs quan sigui necessari
- consultors per accedir als atributs
- operacions de lectura i escriptura

La [segona tasca](#) serà escriure una aplicació simple que permeti gestionar una competició de robots com la que acabem de descriure. L'aplicació haurà de llegir primer les dades d'una competició amb els seus equips i missions, i després oferirà la possibilitat de provar les operacions següents:

1. Mostrar les dades completes de la competició (data, lloc i equips inscrits) [opció -1].
2. Inscriure un nou equip a la competició [opció -2].



3. Modificar un equip de la competició [opció -3].
4. Consultar un equip de la competició [opció -4].
5. Consultar el nombre d'equips inscrits a la competició [opció -5].
6. Consultar l'equip guanyador de la competició [opció -6].
7. Consultar la missió que més punts ha donat als equips de la competició [opció -7].
8. Mostrar les dades finals de la competició (data, lloc i només els equips amb puntuació **ordenats de major a menor puntuació**) [opció -8].
9. Sortir del programa [opció -9].

Recordeu tenir en compte les especificacions que heu fet. En cridar un mètode, comproveu primer si se satisfà la **Pre** i, en cas contrari, decidiu què ha de fer l'aplicació per tractar aquesta situació.

Atès que encara no heu implementat les classes, aquesta aplicació farà ús de les especificacions anteriors però, fins que no les implementeu, no podreu executar l'aplicació. Per tant, la **tercera tasca** serà implementar les classes i generar l'executable de l'aplicació.

La **quarta tasca** serà provar l'executable amb els jocs de prova que us facilitem.

2 Exemples d'algunes de les operacions demanades

1. Mostrar les dades completes de la competició (data, lloc i equips inscrits) [opció -1].

```
Competicio: 08/02/2020, EEBE-UPC
Equips (ordre d'inscripcio):
equip1 | 105 | (missio7, gronxador, 20)
equip2 | 120 | (missio8, muntacarregues, 20)
equip5 | NP |
equip3 | NP |
equip4 | 45 | (missio6, embus_de_transit, 10)
equip6 | 110 | (missio2, grua, 20)
```

Els mostren els equips per ordre d'inscripció. Per cada equip es mostra el seu nom, puntuació i, si té puntuació, la missió amb què ha obtingut més punts. Quan un equip no té puntuació aquesta s'indica amb NP.



2. Consultar l'equip guanyador de la competició [opció -6].

```
Equip guanyador:  
equip2 | 120 | (missio8, muntacarregues, 20)
```

3. Consultar la missió que més punts ha donat als equips de la competició [opció -7].

```
Missio que mes punts ha donat als equips participants:  
(missio2, grua, 20)
```

4. Mostrar les dades finals de la competició (data, lloc i només els equips amb puntuació **ordenats de major a menor puntuació**) [opció -8].

```
Equips (resultats finals):  
equip2 | 120 | (missio8, muntacarregues, 20)  
equip6 | 110 | (missio2, grua, 20)  
equip1 | 105 | (missio7, gronxador, 20)  
equip4 | 45 | (missio6, embus_de_transit, 10)
```

3 Comentaris pel lliurament

- El codi de cada mètode ha d'estar comentat descrivint en quines condicions és aplicable (precondició) i quin és el resultat final (postcondició).
- Per definir les classes **Missio**, **Equip** i **Competicio** no podeu fer servir classes diferents a les que heu vist fins ara al laboratori o a teoria.
- Disposareu de diversos jocs de prova públics. Per cada un d'ells hi haurà un arxiu d'entrada i un altre amb la sortida corresponent. Heu de respectar el format de les entrades i les sortides que apareixen en aquests arxius.
- El que haureu de lliurar és un arxiu comprimit (.zip o .rar) amb tots els arxius *.hpp i *.cpp que siguin necessaris per compilar i muntar el vostre programa (NOMÉS aquests). Dins d'**Atenea** trobareu una Tasca on penjar el vostre arxiu. No s'acceptaran treballs enviats fora de termini ni enviats de cap altra manera.

Data límit: diumenge 15 de març de 2020 a les 23 h 59 min

- Per valorar l'activitat es tindran en compte tant l'especificació com la implementació així com la superació de diverses execucions.



- Per aquesta activitat, heu de formar equips de **dues persones** com a màxim.
- En el cas de detecció de còpia entre dos o més equips, la normativa de l'EPSEVG determina que la qualificació tant de l'activitat com de l'assignatura serà de SUSPENS (0) per totes les persones integrants de tots els equips.

Les accions irregulars que poden conduir a una variació significativa de la qualificació d'un o més estudiants constitueixen una realització fraudulenta d'un acte d'avaluació. Aquesta acció comporta la qualificació descriptiva de suspens i numèrica de 0 de l'acte d'avaluació i de l'assignatura, sense perjudici del procés disciplinari que es pugui derivar com a conseqüència dels actes realitzats.