



## DAM2/DAW2 – Pe1 2020-21

### Enunciat

Implementeu el diagrama de classes de l'esquema de l'[annex A](#) amb els atributs i mètodes indicats. Adjunt a l'enunciat trobareu un programa amb dades de prova «**MainPe1Examen**» que quan acabeu hauria de generar una sortida com la que es mostra a l'[annex B](#).

### Consideracions

- Respecteu **els noms i les definicions** de les classes, interfícies, atributs, mètodes i paràmetres de l'esquema.
- Cal que preueu atenció a:
  - Les **relacions** entre les classes i/o interfícies de l'esquema: associacions, herències, implementacions
  - Els diferents **modificadors** dels atributs i mètodes («private», «public», «protected», «static»)
  - **Formats** que aportin altra informació rellevant: cursiva, subratllat.
  - **Valors** de les constants
- Els «**getters/setters**» no es mostren a l'esquema per simplificar però mentre no s'indiqui el contrari cal crear aquests mètodes per a tots els atributs que no siguin accessibles públicament.
- Podeu afegir **noves constants** públiques o privades a les classes sempre que us calgui. També podeu afegir altres **mètodes privats** (no públics) que us puguin ser d'utilitat.
- Finalment cal que garantiu en la mesura del possible la **coherència de les dades** afegint les validacions que cregueu oportunes. Tant alhora de crear les instàncies com al modificar-les.

### Requeriments

#### Interfície «Avaluable»

<code>esAvaluable(): boolean</code>	Indica si compleix els requeriments per ser avaluable.
<code>getTitol(): String</code>	Retorna un text amb el títol del que s'avalua.
<code>getPuntuacio(): double</code>	Retorna la puntuació del que s'avalua.
<code>getEnunciat(): String</code>	Retorna un text amb l'enunciat del que s'avalua.

#### Classe «UnitatFormativa»

- «cicle»: nom del cicle formatiu, no pot ser nul (substituir per "").
- «modul»: nom del mòdul, no pot ser nul (substituir per "").
- «num»: número d'unitat formativa, no pot ser inferior a 1 (substituir per 1).
- «titol»: títol de la Unitat Formativa, no pot ser nul (substituir per "").
- «hores»: hores lectives de la unitat formativa, no pot ser inferior a 10 (substituir per 10).

<code>UnitatFormativa(cicle, modul, num, titol, hores)</code>	Constructor, inicialitza els atributs de la instància amb els valors indicats per paràmetre.
---	--



### Classe abstracte «Pregunta»

- «PUNTUACIO\_DEFECTE» (constant): puntuació per defecte de les preguntes. Valor 1.0.
- «text»: text amb l'enunciat de la pregunta, no pot ser nul (substituir per "").
- «puntuacio»: valor de la pregunta respecte la puntuació total de l'examen, no pot ser inferior a 0 (substituir per 1).

`Pregunta(text, puntuacio)`

Constructor, inicialitza la Pregunta amb el text i la puntuació indicats per paràmetre.

`Pregunta(text)`

Constructor, inicialitza la Pregunta amb el text indicat per paràmetre i una puntuació per defecte amb valor 1.

`+getEnunciatPregunta(num): String`

Mètode abstracte. Les implementacions han de retornar un text amb la numeració de la pregunta «num» indicada per paràmetre, la **puntuació** i l'**enunciat de la pregunta**.

Les implementacions concretes per les classes «PreguntaOberta» i «PreguntaOpcions» es poden consultar a l'[annex B](#).

### Classe «PreguntaOberta» hereta «Pregunta»

`PreguntaOberta(text, puntuacio)`

Constructor, inicialitza la Pregunta Oberta amb el text i la puntuació indicats per paràmetre.

`PreguntaOberta(text)`

Constructor, inicialitza la Pregunta Oberta amb el text indicat per paràmetre i una puntuació per defecte amb valor 1.

`+getEnunciatPregunta(num): String`

Retorna un text amb la numeració de la pregunta «num» indicada per paràmetre, la **puntuació**, l'**enunciat de la pregunta** i un **requadre** per omplir la resposta.

Es poden consultar exemples a l'[annex B](#).

### Classe «PreguntaOpcions» hereta «Pregunta»

- «opcions»: vector que conté la llista d'opcions disponibles per contestar la pregunta, no pot ser nul (substituir per un vector sense cap element «new String[0] ó new String[]{}»).

`PreguntaOpcions(text, puntuacio, opcions)`

Constructor, inicialitza la Pregunta amb opcions amb el text, la puntuació i la llista d'opcions indicats per paràmetre.

`PreguntaOpcions(text, opcions)`

Constructor, inicialitza la Pregunta amb opcions amb el text i la llista d'opcions indicats per paràmetre i una puntuació per defecte amb valor 1.

`+getEnunciatPregunta(num): String`

Retorna un text amb la numeració de la pregunta «num» indicada per paràmetre, la **puntuació**, l'**enunciat de la pregunta** i la llista d'opcions. Al costat de cada opció es mostra una casella per poder escollir-la.

Es poden consultar exemples a l'[annex B](#).



## Classe «Examen» implementa «Avaluable»

- «MAX\_PREGUNTES» (constant): mida del vector de preguntes. Valor 7.
- «AMPLE\_ENUNCIAT» (constant): ample o mida en caràcters total de l'enunciat de l'examen. Valor 70.
- «AMPLE\_PUNTUACIO» (constant): ample o mida en caràcters de la puntuació que es mostra a la part superior dreta de l'enunciat de l'examen. Valor 15.
- «unitat»: unitat formativa que avalua l'examen, podem suposar que no serà nul en cap cas.
- «preguntes»: vector amb les preguntes de l'examen. La mida d'aquest vector està indicada a la constant «MAX\_PREGUNTES».

### +Examen(unitat)

Constructor, inicialitza l'examen de la unitat indicada per paràmetre i inicialitza el vector de preguntes a la mida establerta per la constant «MAX\_PREGUNTES».

### +afegirPreguntaOberta(text,punt): boolean

Si el vector de preguntes està ple, el text és nul o la puntuació no és superior a 0 retorna «fals».

En cas contrari **afegeix una nova instància de «PreguntaOberta»** al vector «preguntes» a la primera posició disponible i retorna «cert».

### +afegirPreguntaOpcions(text, punt, ops): boolean

Si el vector de preguntes està ple, el text és nul, la puntuació no és superior a 0 o la llista d'opcions és nul·la retorna «fals».

En cas contrari **afegeix una nova instància de «PreguntaOpcions»** al vector «preguntes» a la primera posició disponible i retorna «cert».

### +esborrarPregunta(num): boolean

Si el número indicat com a paràmetre no té un valor entre 1 i el nombre màxim de preguntes retorna fals.

En cas contrari **actualitza a nul la posició «num - 1»** del vector de «preguntes» i retorna «cert».

### +esAvaluable(): boolean

Un examen és avaluable si la seva **puntuació és superior a 0**.

### +getTitol(): String

Retorna un text amb el títol de l'examen que inclou el **nom del mòdul** i de la **Unitat Formativa** que avalua.

### +getPuntuacio(): double

Retorna la **suma de les puntuacions** de totes les preguntes que es troben al vector «preguntes».

### +getEnunciat(): String

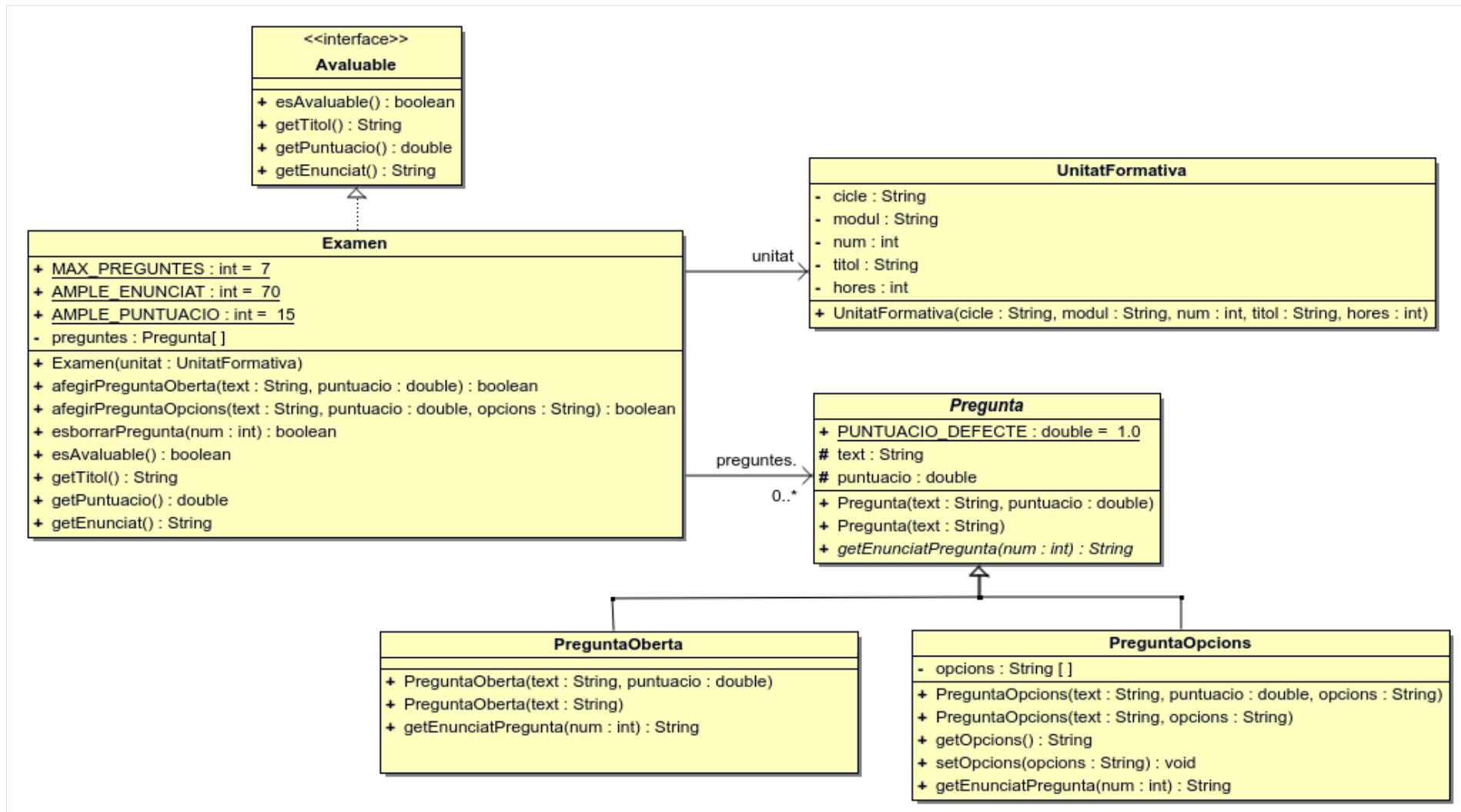
Retorna un **text amb l'enunciat** que inclou totes les preguntes de l'examen numerades a partir de 1 i en l'ordre que es troben al vector «preguntes».

Per obtenir l'enunciat de cada pregunta cal fer servir el mètode «**Pregunta#getEnunciatPregunta(num)**»

Es pot consultar un exemple d'enunciat l'[annex B](#).



## Annex A. Diagrama de classes UML





## Annex B. Extracte de la sortida del programa «MainPe1Examen»

1. "pe1Uf1M7" no és avaluable
2. És correcte!
3. Enunciat "pe1Uf4M3"

##### Examen M03. Programació #####

UF4: Programació orientada a objectes (P00). Fonaments

ENUNCIAT

10,0 PUNTS

Pregunta 1. (1,5 pts) Explica què és una classe abstracte, quines característiques té i les diferències respecte una classe normal.

Pregunta 2. (2,5 pts) Proposa el diagrama UML complet d'una classe que representi una persona

Pregunta 3. (1,0 pts) Marca la resposta correcta. Exemples de tipus de dades primitives poden ser:

- ☐ int, double, String.....
- ☐ char, double, float.....
- ☐ Char, Float, Boolean.....

Pregunta 4. (1,0 pts) Marca la resposta correcta. La millor manera per comparar el text "Hola" amb la variable de tipus String hola és:

- ☐ "Hola" == hola.....
- ☐ hola.equals("Hola").....
- ☐ "Hola".equals(hola).....
- ☐ Cap de les anteriors.....

Pregunta 5. (1,0 pts) Marca la resposta correcta. L'ordre correcte dels elements dins una classe és:

- ☐ atributs > constructor > getters/setters > altres mètodes.....
- ☐ atributs > getters/setters > constructor > altres mètodes.....
- ☐ atributs > altres mètodes > constructor > getters/setters.....
- ☐ altres mètodes > getters/setters > constructor > atributs.....

...



Generalitat de Catalunya  
Departament d'Ensenyament  
INS Marianao

Cicle Formatiu de Grau Superior  
Desenvolupament d'Aplicacions Multiplataforma  
Desenvolupament d'Aplicacions Web



Generalitat de Catalunya  
Departament d'Ensenyament  
INS Marianao

Cicle Formatiu de Grau Superior  
Desenvolupament d'Aplicacions Multiplataforma  
Desenvolupament d'Aplicacions Web