

## TERCERA PRÀCTICA DE PROGRAMACIÓ

### Setmana de la ciència a la URV

#### Presentació del problema

Al mes de novembre la URV, juntament amb altres entitats i institucions, organitza la setmana de la ciència. Aquest any va del 10 al 19. L'objectiu és oferir activitats que fomentin l'interès per la ciència i el coneixement. Volem dissenyar una aplicació que ens permeti registrar dades de les activitats que s'han fet i l'èxit de participació que han tingut.



#### Requeriments d'informació

El sistema de gestió ha de controlar les entitats que han organitzat aquestes activitats. Parlem d'entitats tant per referir-nos a institucions o qualsevol altre organisme. També ha de controlar la informació de les diferents activitats, que seran de tres tipus: visites, tallers i xerrades. En els tallers, els usuaris s'han hagut de registrar per poder-hi accedir, i per tant, volem guardar les seves dades personals i les reserves. A continuació detallem més la informació.

#### Entitats

De cadascuna, en guardarem el nom (que l'identifica), telèfon i correu electrònic de contacte. Tindrem una llista d'entitats.

#### Activitats: visites, tallers i xerrades

Totes les activitats tenen un codi que els identifica, que es genera automàticament a partir de les 3 primeres lletres del nom de l'entitat, i seguides d'un número (començant per 100). De totes les activitats es guarda també el nom, el lloc, el codi postal, i el dia en que es fa (el mes i l'any no cal, ja assumim que són de novembre de 2023). També haurem de poder indicar quina entitat l'ha creat. Per a que hi hagi varietat, les activitats no es repeteixen mai, és a dir, cada activitat ha de guardar un sol dia.

Segons el tipus d'activitat, haurem de guardar alguna informació addicional:

- Les visites consisteixen en obrir algun edifici o espai de la ciutat per accés lliure i gratuït. Com unes "portes obertes". Algunes ofereixen audioguia i altres no. Algunes estan adaptades per persones cegues.

- Els tallers es fan en una hora concreta del dia, i tenen una durada determinada. Tenen també una capacitat fixada, i els usuaris s'hi ha de registrar.
- Les xerrades es fan sempre a la mateixa hora, les 17 hores. Volem guardar el nom de la persona que fa la xerrada.

Tindrem una llista d'activitats.

## Usuaris

Per poder participar en un taller s'han d'inscriure prèviament. Un usuari es dona d'alta al sistema amb un àlies que l'identifica i és diferent per a tots els usuaris. De l'usuari guardem aquest àlies, la seva adreça de correu electrònic i el codi postal d'on viu. Un usuari es pot apuntar a diferents tallers.

Tindrem la llista d'usuaris.

## Reserves dels usuaris als tallers

Quan un usuari demana anar a un taller es comprova que en el taller encara hi ha places disponibles i, si és el cas, es genera una instància reserva amb un codi. De cada reserva es guarda, el codi de la reserva, l'usuari que fa la petició i el codi del taller al què vol assistir. El taller ha de registrar d'alguna forma que ja té un nou assistent i per tant, una plaça disponible menys.

Quan l'usuari ha fet el taller, podrà valorar el seu nivell de satisfacció amb un valor en el rang [0-10]. La valoració es guarda en la reserva i s'aplica al taller on es guarda la suma de puntuacions que ha rebut el taller, junt amb el nombre de persones que l'han puntuat.

Tindrem la llista de reserves.

## Requeriments no funcionals del disseny

Tingueu en compte que, a més de les classes corresponents als elements que s'han acabat de descriure, heu d'implementar les classes de les **l·listes** que els emmagatzemen. Es recomana que en cada llista els elements estiguin ordenats per l'atribut més adient segons les cerques que es demanen (veure més avall). **ATENCIÓ:** No es poden usar les classes de Java per guardar les l·listes (com ArrayList, o altres). Recordeu que els mètodes de la llista no han de retornar sempre String, sinó que han de retornar les dades del tipus que correspongui, i serà la part de visualització la que s'ocupi de mostrar-les com calgui.

Les classes que emmagatzemen la informació de l'aplicació es faran persistents mitjançant **fitxers**. A cada nova execució del programa, es carregarà automàticament tota la informació dels fitxers i es crearan i inicialitzaran els objectes de les classes corresponents. Durant l'execució de l'aplicació, les dades s'aniran modificant sobre els objectes de memòria com a resultat de les funcionalitats demanades pels usuaris. Quan es vulgui tancar l'aplicació, es demanarà a l'usuari si vol guardar tota la informació que hi ha als objectes de memòria als fitxers, sobreescrivint el contingut que tenien (aquesta opció està pensada per a facilitar els jocs de proves).

Les dades de les reserves es guardaran en un **fitxer serialitzat**. La resta d'informació (entitats, activitats i usuaris) es guardarà en **fitxers de text**. Tots els fitxers de text han de seguir un format on les dades de cada instància han d'estar en una línia amb els diferents atributs separats pel caràcter ';'. A la documentació final s'ha d'explicar el format dels fitxers utilitzats, posant algun exemple. També cal incloure els fitxers de prova al projecte que s'entregarà. Els fitxers de prova han de ser suficientment complets per poder comprovar les diferents funcionalitats demanades. També hi ha d'haver un nombre raonable de dades que garanteixin que heu provat suficientment la vostra aplicació.

S'haurà d'afegir el **control d'excepcions**. S'han de tenir en compte les excepcions predefinides en Java que es puguin donar. També s'ha de definir alguna excepció pròpia que sigui coherent amb els requeriments de la pràctica. La **documentació** també ha d'explicar detalladament tots els errors que es controlen, on i amb quina excepció (predefinida o pròpia).

La interfície gràfica que es demana més avall s'ha d'implementar en un package a banda, separada de les classes de dades. Aquesta interfície s'implementarà usant les classes SWING i es programarà des de zero (no es pot usar cap eina d'implementació d'interfícies automàtica).

No es poden usar eines de generació de codi automàtic, en cap de les parts de la pràctica.

## Requeriments funcionals

Es demana realitzar dues aplicacions, una amb entrada i sortida per consola, i una altra on l'entrada i sortida es farà mitjançant interfície gràfica. A continuació es detallen les funcionalitats que ha d'oferir cadascuna d'elles. **Important:** totes les aplicacions treballaran sobre la mateixa informació, és a dir, les mateixes dades carregades dels mateixos fitxers.

### Aplicació amb entrada/sortida per consola

L'aplicació carrega la informació dels fitxers i manega l'entrada/sortida per la consola. Abans de mostrar el menú amb les funcionalitats es llegiran les dades dels fitxers i s'inicialitzarà les estructures llistes amb tota la informació. Les opcions de menú han de ser:

1. Mostrar les dades de qualsevol llista que tingueu definida.
2. Obtenir i mostrar la llista d'activitats que ofereix una entitat concreta.
3. Obtenir i mostrar la llista de les activitats que es duen a terme en un dia indicat per teclat.
4. Obtenir i mostrar la llista dels tallers que tenen places disponibles.
5. Afegir una nova activitat
6. Registrar la petició d'un usuari per reservar un taller.
7. Mostrar els usuaris que s'han apuntat a un taller.
8. Calcular l'usuari que s'ha apuntat a més tallers.
9. Registrar la nota que un usuari que s'ha apuntat a un taller li dona un cop s'ha fet.
10. Calcular la nota mitja que ha rebut un taller.

11. Quin és el taller que ha tingut més èxit? Calcularem l'èxit segons el taller que ha tingut una ocupació més alta en proporció a les places que oferia.
12. Obtenir i mostrar les dades de la llista de visites ofertes per una entitat (per teclat s'indicarà si es vol audioguia o si cal que estigui adaptada per persones cegues).
13. Mostrar les dades de les xerrades que farà una persona concreta.
14. Donar de baixa un taller sempre que no hi hagi usuaris apuntats.
15. Sortir de l'aplicació.

## **Aplicació amb entrada/sortida gràfica**

L'aplicació carrega la mateixa informació dels fitxers que es fa servir en la versió per consola. Aquesta aplicació té un conjunt molt més reduït de funcionalitats que ofereix mitjançant una interfície gràfica. Cal implementar-ho amb les eines del paquet SWING, sense usar cap generador de codi automàtic.

Volem mostrar gràficament les activitats que es fan en un període de 10 dies, per tant tindrem 10 botons a la finestra. A cada botó hi pintarem el número d'activitats previstes en aquell dia. Si cliquem sobre el botó que es mostri informació de la llista d'accions del dia.

La finestra tindrà 3 botons per filtrar segons tipus d'activitat. Quan un dels tipus estigui seleccionat, els botons només mostraran activitats d'aquell tipus. Estaran en vermell si no hi ha cap activitat del tipus indicat en un dia concret.

## **Instruccions per al desenvolupament de la pràctica**

- **Aquesta pràctica es treballa en equip de quatre persones.** A l'espai moodle compartit entre Programació i PSI teniu una activitat per apuntar-vos a un dels equips. Treballar en equip vol dir repartir-se les tasques necessàries per resoldre tot el treball, no vol dir fer-ho tot entre tots. Així doncs recomanem fer primer una lectura global de la pràctica, analitzar el conjunt de classes que són necessàries i repartir-vos la feina. A més, cada persona és responsable del bon disseny i funcionament de la seva part. Cada equip tindreu assignat un o dos supervisors de 4rt de GEI. Ells us ajudaran a fer el disseny del programa, a repartir la feina i a treballar de forma coordinada. Cal seguir les seves indicacions però ells no programen res, ni resolen dubtes tècnics. Consulteu als professors de Programació en cas de dubtes de com implementar alguna part.
- **Cada equip tindrà assignat un professor/a de laboratori.** Ell/a revisarà la distribució de la feina que us feu entre els membres de l'equip i el disseny de classes inicial. Comuniqueu al professorat també si hi ha algun problema amb l'equip (baixes de membres, desajustos en la planificació, dificultats amb el supervisor/a, etc).
- A més, per a validar el treball i assegurar que heu plantejat un bon disseny de classes i esteu abordant el problema de forma correcta us **proposem una entrega voluntària el dia 15 de desembre de 2023**. L'entrega ha de servir per a validar a grans trets que l'estructura de classes, la forma d'utilitzar-les, ... és correcta. En aquesta data no es demana tot el codi, sobretot per a la part d'interfície gràfica no s'espera res, encara. Malgrat ser una entrega voluntària es gratificarà

amb fins a 1 punt extra a la nota de la pràctica que s'avaluarà al gener. El valor de la nota extra dependrà de que hi hagi un mínim fet al que s'entrega.

- La comunicació bàsica entre els membres de l'equip s'ha de fer via el fòrum d'equip que tindreu al moodle en l'espai compartit. Valorarem el vostre treball continu sobre el contingut d'aquest fòrum.
- El supervisor/a de l'equip farà una acta de cada reunió al mateix fòrum d'equip. Cal que la consulteu per comprovar els acords presos i cal que compliu el que heu acordat.
- És molt important la validació de cada classe per separat, és a dir, fer un seguit de proves unitàries per classe. Els codis per a valorar aquestes classes es poden incloure en l'entrega final posant-los en un Package a banda.
- Cal lliurar també un informe final que és part de la nota. A més, avaluarem la competència C4 (escriptura en català o castellà). L'informe ha de seguir el format i contingut indicats a les instruccions que trobareu al Moodle.
- Heu de crear el *Java Project* amb el vostre número d'equip, **numEquip\_nomPractica**. Recordeu estructurar el codi en diferents packages, com a mínim, un per a l'aplicació de consola, un altre per a l'aplicació amb interfície gràfica i un altre per a les classes que guarden les dades.
- Recordeu que totes les classes han d'estar ben documentades i utilitzar els comentaris JavaDoc.
- Lliurareu els codi fonts i l'informe en pdf. Tota aquesta informació es guardarà en un fitxer ZIP i es penjarà a la tasca corresponent del Moodle. El nom del fitxer serà **nomEquip\_nomPractica.zip**
- El termini de lliurament és el **15 de gener de 2024**.
- En les sessions de laboratori de la setmana de classes al gener, del 8/01 al 12/01, hi haurà una **entrevista individual** sobre la pràctica que pot influir en la nota final. L'entrevista és individual però tots els membres de l'equip l'han de fer en la mateixa franja de temps.

## Criteris d'avaluació

- Referents a la implementació:
  - Correctesa en la definició de les classes (tipus dels atributs, modificadors, herència, ...).
  - Correctesa en la definició i implementació dels mètodes (que rebin els paràmetres adients i facin correctament el que es demana, de la forma més senzilla i òptima possible).
  - Manegament correcte de les llistes (recorregut/cerca).
  - Cal manegar les excepcions predefinides de Java i alguna de pròpia.
  - Lectura/escriptura de fitxer de forma senzilla i eficient.
  - El codi que s'implementi ha de ser clar: ha d'estar tabulat, ha d'estar comentat adequadament (cada mètode ha de tenir una capçalera indicant què fa, i quins paràmetres rep) i els noms que useu pels atributs, mètodes, etc han de ser representatius.
  - Ús de packages per separar el codi.
  - Ús correcte de la classe SWING per fer la interfície gràfica.
- Referents a la competència B8/CT4 i CT5:
  - Es valorarà la redacció, estil, correctesa de l'informe de la pràctica.

- Es valorarà la distribució de tasques i treball realitzat de forma individual i de forma conjunta.
- Referents al disseny i joc de proves i al treball personal de cada membre:
  - Es valorarà el contingut de l'informe i els jocs de prova entregats.
- Altres consideracions:
  - No s'acceptaran pràctiques que no funcionin o no facin el que demana l'enunciat.
  - Reviseu que el codi compili sense advertiments.
  - En cas de detectar-se dues o més pràctiques iguals o molt similars es posarà una nota de 0 a tots els alumnes implicats.