

MÒDUL 3: PROGRAMACIÓ (UF4, UF5 i UF6)

PRÀCTICA UF5 (PART 1)

Continuem amb el programa de gestió dels parcs d'atraccions de la UF4.

En aquesta pràctica, a partir del projecte NetBeans com a base per aquest exercici, què trobareu en el Moodle, heu de desenvolupar i/o completar les classes per aconseguir implementar els requeriments que s'expliquen a continuació.

Per aquesta pràctica, els requeriments s'agrupen en tres grans blocs:

- Pel que fa a excepcions
 - es controlarà que l'usuari doni opcions numèriques als menús ✓
 - es controlarà no sobrepassar el màxim d'elements dels vectors
 - s'afegirà el llançament d'excepcions pròpies per controlar errors o validacions
- Pel que fa a col.leccions
 - conservareu els arrays a Aplicacio i a ParcAtraccions
 - canviareu l'array de Zona per un ArrayList
- Pel que fa a XML
 - fareu persistent el parc d'atraccions actual, és a dir, podreu desar a un fitxer totes les dades del parc d'atraccions actual , amb els seus coordinadors, personal de manteniment, atraccions i zones, i després es podrà recuperar des de el disc creant els objectes adients i les seves propietats. No farem persistent els registres.

Abans de descriure els requeriments detallats per a cada classe, heu de tenir en compte les següents consideracions:

- L'array de parcs d'atraccions a Aplicacio el posem d'un element, és a dir, per simplificar el programa, només es gestionarà un parc d'atraccions.
- Al programa teniu una interfície amb nom Element, i a la gestió de XML necessitareu la classe Element del paquet XOM. Heu de tenir cura de confondre ambdues classes.
- S'ha afegit el paquet «persistencia» amb les classes / interfícies GestorPersistencia i ProveedorPersistencia que no s'han de modificar. En aquest mateix paquet també teniu la classe GestorXML que és la que haureu de modificar / desenvolupar.
- Aquest esquema de classes al paquet «persistencia» s'ha fet així, perquè al llarg del curs, en pràctiques posteriors, haureu d'afegir altres classes de persistència (sobre bases de dades relacionals i sobre bases de dades orientades a objectes)

| | |
|--------------|----------------|
| Document: | Data: 17/12/15 |
| Pràctica UF5 | Pàgina 1 de 4 |

- És molt important desenvolupar en l'ordre que s'us mostra en la taula següent, la de requeriments detallats.

Els requeriments detallats que heu de desenvolupar són els de la taula següent:

| Classe | Bloc | Mètode | Requeriments detallats a desenvolupar |
|------------------------|------------------|--|---|
| ParcAtraccionsExcepcio | Excepcions | getMessage() | Com podeu veure en el codi, aquesta classe exten d'Exception, per tant, getMessage() és el mètode d'Exception. Heu de sobreescrivre el mètode per mostrar el codi de causa del missatge i el text del missatge. |
| Zona | Col·leccions | --- | Canvieu l'atribut elements per un ArrayList Elimineu la propietat comptaElements |
| | | get's i set's | Canvieu / elimineu els get's i set's per ser coherents amb el canvi / eliminació dels atributs anteriors |
| | | afegeixElementZona | Canvieu per gestionar l'ArrayList en lloc de l'array. |
| | | mostraElement | Canvieu per gestionar l'ArrayList en lloc de l'array. El recorregut s'ha de fer amb un Iterator. |
| ParcAtraccions | Excepcions | seleccionaCoordinador(String pNif) | public Integer seleccionaCoordinador(String pNif) throws ParcAtraccionsExcepcio { Nou mètode que donat un nif per paràmetre (pNif) ha de buscar-lo a l'array d'elements de Granja (elements) i si el troba, ha de llançar l'excepció GranjaExcepcio amb codi causa = "3" Si no troba el nif, llavors es retorna un integer → -1 Per la pròpia declaració del mètode, es veu que la gestió de l'excepció serà delegada al mètode que el crida. |
| | Excepcions i XML | nouCoordinador | public void nouCoordinador(Coordinador coordinador) Aquest mètode és el que serveix per inserir un element de tipus Coordinador a l'array d'elements del ParcAtraccions. S'ha de canviar per dos motius: a) Ha d'acceptar el paràmetre Coordinador. Si el paràmetre és null, es fa com fins ara, és a dir s'inserirà Coordinador.nouCoordinador() Si el coordinador del paràmetre no és null, llavors és aquest directament el que s'inserirà (no es crearà cap coordinador nou perquè ja està creat, és el què es passa per paràmetre) b) Ha de comprovar que el nif no existeixi Abans d'inserir el nou coordinador o el coordinador del paràmetre a l'array, s'ha de comprovar que el nif no estigui ja donat d'alta. Per fer això, heu de fer servir el mètode seleccionaCoordinador. Com es crida a seleccionaCoordinador, si s'ha produït l'excepció, llavors s'ha de capturar en aquest mètode i imprimir el missatge per pantalla. |
| | | novaPersonaManteniment, novaAtraccio, novaZona | Heu de fer el mateix que a nouCoordinador però amb l'element corresponent. |
| | XML | Nou constructor | Heu de fer un nou constructor (i conservar el que ja teniu) public ParcAtraccions(Integer codi, String nom, String adreça) La diferència és el paràmetre codi. En aquest cas, com que el codi ve donat, llavors no s'ha d'incrementar la propietat comptaCodi. |
| | | seleccionaElement | public Integer seleccionaElement(Integer tipusElement, String codi) S'ha afegit el paràmetre String codi. Si el codi ve donat al paràmetre, llavors no s'ha de demanar per consola i es farà servir aquest (entenent que codi pot ser un nif o un nom, depèn de l'element) |

| Classe | Bloc | Mètode | Requeriments detallats a desenvolupar |
|---------------------|------------|--------------------|--|
| | | afegeixElementZona | Heu de canviar les crides a seleccionaElement passant codi = "" |
| MARTIN Aplicacio | Excepcions | main | Heu de gestionar les excepcions de tipus ParcAtraccionsExcepcio què es poden produir el menú principal. Un cop capturades les excepcions heu de mostrar el missatge i codi associat a l'excepció, mitjançant el mètode getMessage() de la classe ParcAtraccionsExcepcio. |
| | | menuPrincipal() | Heu de gestionar les excepcions de tipus ParcAtraccionsExcepcio què es poden produir en el menú parc d'atraccions. Un cop capturades les excepcions heu de mostrar el missatge i codi associat a l'excepció, mitjançant el mètode getMessage() de la classe ParcAtraccionsExcepcio. |
| | General | --- | S'han afegit nous atributs. No heu de canviar-los. |
| | | menuElement | Ha de cridar als mètodes nouXXX amb un null per paràmetre Ha de cridar a seleccionaElement afegint el paràmetre codi. En aquest cas, serà "" |
| | | menuZones | Ha de cridar a novaZona amb un null |
| | XML | menuParcAtraccions | S'ha tret l'opció d'afegir registre. Hi ha noves opcions (carregar i desar parc d'atraccions), però en aquest mètode no heu de modificar res. |
| MARC GestorXML | XML | General | L'objectiu és que aquesta classe pugui desar el parc d'atraccions al fitxer "parcAtraccions.xml" i després pugui recuperar un parc d'atraccions des d'aquest mateix fitxer. El nom del fitxer (sense l'extensió) és l'atribut FITXER d'Aplicacio. L'extensió (".xml") es posarà directament en GestorXML. A l'arrel del projecte que hi ha penjat al Moodle, teniu un exemple d'aquest fitxer. Aquest exemple mostra un Parc d'atraccions amb dos coordinadors, dos persones de manteniment, dos atraccions i una Zona. La Zona a la vegada contenen coordinadors, persones de manteniment i atraccions del parc d'atraccions. |
| | | construeixModel | L'objectiu es llegir els atributs de l'objecte ParcAtraccions passat per paràmetre per construir un model (document XML) sobre el Document doc (atribut de GestorXML). Com es veu a l'exemple, l'arrel del document XML és "parcAtraccions" i heu d'afegir-ne els valors de codi, nom i adreça com atributs. Aquesta arrel, l'heu d'afegir a doc. Un cop fet això, heu de recórrer l'array elements de ParcAtraccions i per a cada element, afegir un fill a doc. Cada fill tindrà com atributs els atributs de l'objecte (nif, nom, ...) Si es tracta d'una Zona, a més, heu d'afegir fills addicionals amb els valors dels elements d'aquesta zona (els seus coordinadors, personal de manteniment i atraccions) |
| | | desarModel | No heu de fer res. |
| | | carregarFitxer | No heu de fer res. |
| | | fitxerGranja | El mètode carregarFitxer llegeix el fitxer del disc i el carrega sobre l'atribut doc de GestorXML. L'objectiu és llegir el doc per omplir els atributs de ParcAtraccions (i la resta d'objectes) Per fer això, com l'arrel conté els atributs del parc d'atraccions, al fer Element arrel = doc.getDocument(); ja podeu crear l'objecte ParcAtraccions. Un cop fet això, heu de recórrer el doc i per cada fill, haureu d'afegir un element a l'array elements de ParcAtraccions (nouXXX(...)) Si el fill (del doc) que s'ha llegit és un Zona, recordeu que a més de fer el novaZona per afegir-lo a ParcAtraccions, també haureu d'afegir en aquesta zona els seus coordinadors, personal de manteniment i atraccions. |

La valoració d'aquest exercici es basarà en els criteris següents:

- Qualitat del codi (el codi està ben comentat, es llegeix bé, s'han donat noms correctes a les variables, no utilitza més recursos del necessari, i fa bon ús de l'orientació a objectes): 2 punts
- Correcte funcionament del desenvolupament de les següents classes i interfícies:
 - ParcAtraccionsExcepcio → 0,5 punts
 - Zona → 1,5 punts
 - ParcAtraccions → 2 punts
 - Aplicacio → 1,5 punts
 - GestorXML → 2,5 punts

La pràctica l'heu de fer en grups de 2 persones.

Es corregirà el dia **09/02/2016** de manera presencial en les hores de classe.

Teniu temps per lliurar la pràctica fins **dimarts dia 09 de febrer a les 19:00** en l'enllaç que trobareu en el curs del Moodle.

| | |
|------------------|-----------------------|
| Document: | Data: 17/12/15 |
| Pràctica UF5 | Pàgina 4 de 4 |