Programazio Modularra eta Objektuei Bideratutako Orientazioa – 5. Laborategia

Aurrebaldintzak

Ikasleak ondo menperatzen du Eclipse ingurunea, baita Java perspektiba ere. Ataza bat burutzeko, klase ugari behar direnean badaki klaseak sortzen eta bakoitzaren atributuak eta metodoak ondo ezartzen, elementu bakoitzak bakarrik dagokion informazioa kudeatzen duelarik. EMAk eta DMAk desberdintzen ditu. Eta UML diagramen bitartez, klaseen atributu eta metodoak, eta klaseen arteko erlazioak errepresentatzeko gai da. Ikasleak badaki proba programak sortzen Junitak erabiliz.

Helburuak

Laborategi honen helburua, aurreko laborategietan ikasitako kontzeptuak bermatzea du helburu.

Laborategi hau bukatzean, ikasleak jarraian agertzen dena egiteko gai izango da:

- EMAk eta DMAk dituzten ariketak egiteko gai izango da.
- Listak (*ArrayList*) eta iteratzaileen erabilpena menperatzen du.
- Junit-ak errez inplementatzen ditu.

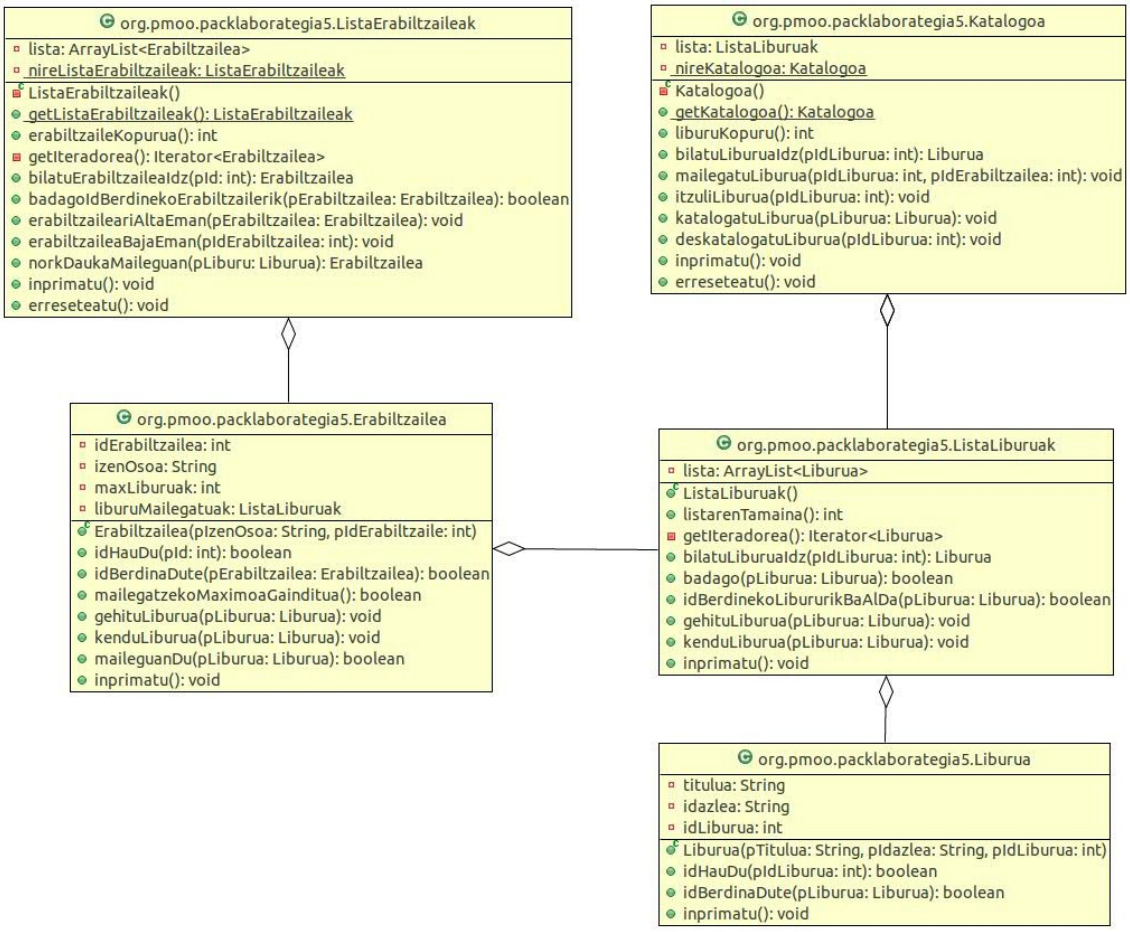
Motibazioa

Laborategi honetan konplexutasuna handitzen da, erabili behar direlako 5 klase.

Laborategi honetan ataza bakar bat aurkezten da, nun 5 elkar-erlazionatutako klaseak erabiliko diren. Diseinuari dagokionez, klase diagrama egitea eskatzen da. Nahiz eta bukaeran adostuko diren, beraz, atazaren muina inplementazioan kokatzen da, eta ez diseinuan. Ala ere, enuntziatua irakurtzean, klaseak identifikatzen hastea gomendatzen da, bukaeran proposatzen diren horiekin alderatuz.

Ataza bakarra: Udal-liburutegi baten kudeaketa

Ataza honen helburua, auzo bateko udal-liburutegi baten kudeaketa gauzatzean datza, liburuen maileguari dagokionez. Horretarako, azpian erakusten den UML klase diagramak jarraitzen duen aplikazioa garatu behar duzue.



1. irudia: Liburutegiaren klase diagrama

Aplikazioak, liburutegiaren erabiltzaileen informazioa kudeatu behar du. Erabiltzaile bakoitzeko, erabiltzailearen identifikadorea, bere izen osoa (izen-abizena), eta maileguan izan ditzakeen liburu kopuru maximoa eta momentuan maileguan dituen liburuen zerrenda.

Aplikazioak, baita ere, liburutegiak dituen liburu guztien zerrenda (Katalogoa) gorde behar du. Liburu bakoitzeko gordeko den informazioa, liburuaren identifikadorea, titulua, eta idazlearen izen osoa (izen-abizena).

Aplikazioak izan behar dituen oinarrizko funtzionalitateak hurrengoak dira:

1) Liburu eta erabiltzaileak erregistratu eta alta/baja ematea.
2) Liburu eta erabiltzaileen bilaketa identifikadore bitartez.
3) Liburuen mailegutza eta itzulketak.
4) Katalogo eta erabiltzaileen informazioa pantailaratzea.

Honako klase hauek garatu behar dira:

Liburua klasea

- **Atributuak:** titulua (String), idazlearen izen osoa (String) eta identifikadorea (int)
- **Metodo eraikitzailea:** behar diren atributuak hasieratzeko parametroak jasoko ditu.
- **Gainontzeko metodoak:**
  - *idHauDu()* parametro gisa jasotzen duen identifikadore berbera duen ala ez adierazteko boolear bat bueltatuko du.
  - *idBerdinaDute()* parametro gisa Liburu bat jasoko du eta identifikadore berbera duten ala ez adierazteko boolear bat bueltatuko du. Metodo honek Liburu bat jasoko du, idHauDu() metodoak berriz identifikadore bat.
  - *inprimatu(),* metodo honek liburuaren atributuak pantailaratuko ditu.

ListaLiburuak klasea

- **Atributua:** ArrayList baten bitartez inplementatzen den liburu zerrenda bat.
- **Metodo eraikitzailea:** ArrayList instantzia hutsa sortuko du eta lista atributuari esleitzen zaio.
- **Gainontzeko metodoak:**
  - *listarenTamaina():* zerrendan dauden liburu kopurua itzuliko du.
  - *getIteradorea():* liburu zerrenda zeharkatzeko iteradore bat itzuliko du.
  - *bilatuLiburuaIdz():* sarrera gisa liburu baten identitatea jasoko du eta liburu hori itzuliko du. Ez bada existitzen null balioa itzuliko du.
  - *badago()* parametro gisa jasotzen duen liburua zerrendan dagoen adierazteko boolear bat bueltatuko du.
  - *IdBerdinekoLibururikBaAlDa():* Boolear bat itzuliko du baldin eta sarrera gisa jasotzen duen liburuaren identifikadorea duen libururik zerrendan existitzen den azaltzeko.

- *gehituLiburua()* parametro gisa jasotzen duen liburua mailegatutako liburuei gehituko die, noski, soilik aurretik ez balego.
- *kenduLiburua()* parametro gisa jasotzen duen liburua zerrendatik ezabatzen du.
- *inprimatu()* zerrendako liburu bakoitzaren informazioa inprimatuko du.

Erabiltzailea klasea

- **Atributuak:** identifikadore bat (int), izen osoa (String), mailegatu ditzaken liburu kopuru maximoa (int) eta maileguan dituen liburuen zerrenda (ListaLiburuak).
- **Metodo eraikitzailea:** parametro gisa jasoko dituen izen osoa eta identifikadorearekin hasieratuko ditu dagokien atributuak. MaxLiburuak defektuzko balio batekin hasieratuko du, hau da, 3, eta liburuMailegatuak zerrenda hutsik hasieratuko du.
- **Gainontzeko metodoak:**
  - *idHauDu()* erabiltzaileak, metodo honek parametro gisa hartzen duen identifikadore berbera duela konprobatzen du, emaitza boolear baten bitartez bueltatuz.
  - *idBerdinaDute()* parametro gisa erabiltzaile bat jasoko du eta identitate berdina duten esango du. Metodo honek erabiltzaile bat jasoko du, aurreko metodoak ordea erabiltzailearen identitatea jasoko du.
  - *mailegatzekoMaximoaGainditua()* erabiltzaileak mailegatu ditzakeen liburu kopurura heldu denean true bueltatuko du metodo honek.
  - *gehituLiburua()* jasotzen duen liburua mailegatutako liburuen zerrendan gehituko du.
  - *kenduLiburua()* jasotzen duen liburua mailegatutako liburuen zerrendatik ezabatuko du.
  - *maileguanDu()* parametro gisa jasotzen duen liburua, bere liburu zerrendan duen adierazteko boolear bat bueltatzen du.
  - *inprimatu()* erabiltzailearen datuak pantailaratzen ditu, eta liburuak maileguan dituen ala ez adieraziko du ere. Liburuak balitu, liburuen informazioa ere pantailaratuko du.

<u>Katalogoa Klasea</u> (EMA)

- **Atributuak:** liburu zerrenda bat (*ListaLiburuak*), eta *nireKatalogoa,* alegia, EMAren instantzia bakarra jasotzeko atributu estatikoa.
- **Metodo eraikitzailea:** zerrenda hasieratuko du.

Gainontzeko metodoak

- *getKatalogoa() Katalogo*-aren instantzia bakarra bueltatzen du, honela edozein klasetik Katalogoaren instantzia bakarra atzipena ahalbidetzen da.
- *liburuKopuru()* Katalogoak dituen liburuen kopurua bueltatzen du.
- *bilatuLiburuaIdz()* Parametro bezala jasotzen duen liburuaren identifikadore bera duen *Katalogoko* liburua bueltatuko du, bestela eta ez balu topatuko null bueltatuko luke. **KONTUZ!!, bilaketa egiteko ez erabili *equals.***
- *mailegatuLiburua()* parametro gisa liburu baten id eta erabiltzaile baten id jasoko ditu, eta id hori duen liburua erabiltzailearen liburu mailegatuen zerrendan gehituko du, baldin eta erabiltzaileak maileguan izan ditzaken liburu kopuru maximora heldu ez bada, edo liburu hori beste erabiltzaile batek ez balu maileguan. Aurreko egoeren batengatik eragiketa burutzea posible izango ez balitz , mezu bat pantailaratuko da egoeraz informatzen.
- *itzuliLiburua(),* parametro gisa jasotzen duen liburuaren identifikadoreaz baliatuz, liburua nork duen asmatu ondoren, erabiltzailearen liburu mailegatuen zerrendatik ezabatuko du liburua.
- *katalogatuLiburua()* parametro gisa jasotzen duen liburua gehituko du Katalogoan, beti katalogatu nahi den liburua ez egotea konprobatu ostean. Horrela balitz, pantailatik mezu bat agertuz.
- *deskatalogatuLiburua()* parametro gisa jasotzen duen identifikadorea duen liburua Katalogotik ezabatzen du, beti liburu hori katalogokoa dela ziurtatu ostean. Horrela balitz, pantailatik mezu bat agertuz. Gainera, liburua mailegatuta EZ dagoela ziurtatu behar da.
- *inprimatu() Katalogoak* dituen liburu kopurua pantailaratuko du, baita liburu bakoitzaren informazioa ere.
- *erreseteatu() Katalogoa-ren* instantzia bakarra erreseteatzen du, hau da, hutsik uzten du. **OHARRA:** Metodo hau jartzen da bakarrik Junit-en egikariketa errazteko, baina aplikazio erreal batean inoiz ez da agertuko eta gutxiago publiko gisa, edonork edonoiz gure *Katagoa* ezabatu dezan ahalbidetuz.

<u>ListaErabiltzaileak klasea</u> (EMA)

- **Atributuak:** ArrayList baten bitartez inplementatutako zerrenda bat, eta *nireListaErabiltzaileak* atributu estatikoa nun klasearen instantzia bakarra gordeko den.
- **Metodo eraikitzailea:** ArrayList instantzia hutsa sortuko du eta lista atributuari esleitzen zaio.
- **Gainontzeko metodoak:**
  - *getListaErabiltzaileak() ListaErabiltzaileak* klasearen instantzia bakarra bueltatzen du, honela edozein klasetik instantzia bakar horren atzipena ahalbidetzen da.
  - *erabiltzaileKopurua()* zerrendak duen erabiltzaileen kopurua bueltatzen du.
  - *getIteradorea()* zerrenda zeharkatzeko iteradorea itzuliko du.
  - *bilatuErabiltzaileaIdz()* parametro gisa jasotzen duen erabiltzailearen identifikadorearekin kointziditzen duen zerrenda barruko erabiltzailea bueltatuko du. Ez balego horrelakorik null bueltatuko du.
  - *badagoIdBerdinekoErabiltzailerik()* boolear bat bueltatuko du, parametro gisa jasotzen duen erabiltzailearen identifikadore bera duen beste erabiltzaileren bat existitzen duen adierazteko.
  - *erabiltzailеariAltaEman()* sarrera gisa erabiltzaile bat jasoko du eta erabiltzaileen zerrendan gehituko du. Beti konprobatuz horrelako erabiltzailerik ez den existitzen jada.
  - *erabiltzaileaBajaEman()* sarrera gisa erabiltzaile baten identitatea jasoko du eta erabiltzailea zerrendatik kenduko du.
  - *norkDaukaMaileguan()* liburu bat jasotzen du parametro gisa eta liburu hori maileguan duen erabiltzailea bueltatuko du. Inork ez balu liburu hori izango, orduan null bueltatuko du.
  - *erreseteatu()* instantzia bakarra erreseteatzen du, hau da, hutsik uzten du. ***OHARRA:*** *Metodo hau jartzen da bakarrik Junit-en egikariketa errezteko, baina aplikazio erreal batean inoiz ez da agertuko eta gutxiago publiko gisa, edonork edonoiz gure instantzia ezabatu dezan ahalbidetuz.*
  - *inprimatu()* zerrendaren erabiltzaile kopurua inprimatzen du, eta gero erabiltzaile bakoitzaren informazioa.