

Osasun-etxea

Munitibarko osasun-etxea kontsulten datuen informatizazio-prozesuan murgilduta dago. Prozesu hori diseinatzeko, zentroak pazienteen informazioa gorde eta kudeatu behar du. Paziente bakoitzerako, izena, abizenak, adina, gizarte-segurantzako zenbakia, helbidea eta bere sintomen zerrenda gordetzen dira. Etorkizunean hainbat funtzionalitate sartzeari pentsatzen denez, sintomen katalogo bat erabiliko da, eta sintoma bakoitzerako, izena eta bere kode numerikoa gordeko dira.

1. E11: Sintoma jakin bat daukaten paziente kopurua kalkulatu

Gure helburua hurrengoa da: sintoma bat sartuta (String formatuan), azken hori daukaten paziente kopurua kalkulatu.

Hurrengoa eskatzen da:

- Osatu ezazu *klase diagraman aipatutako funtzionalitatearekin*. Pazienteen sintomei dagokienez, osagai primitiboak erabilti itzazu sintomak adierazteko.
- bilatuPazienteKop(pSint:String):int* metodoari dagokion *sekuentzia diagrama* osatu UML formalismoaren bidez.

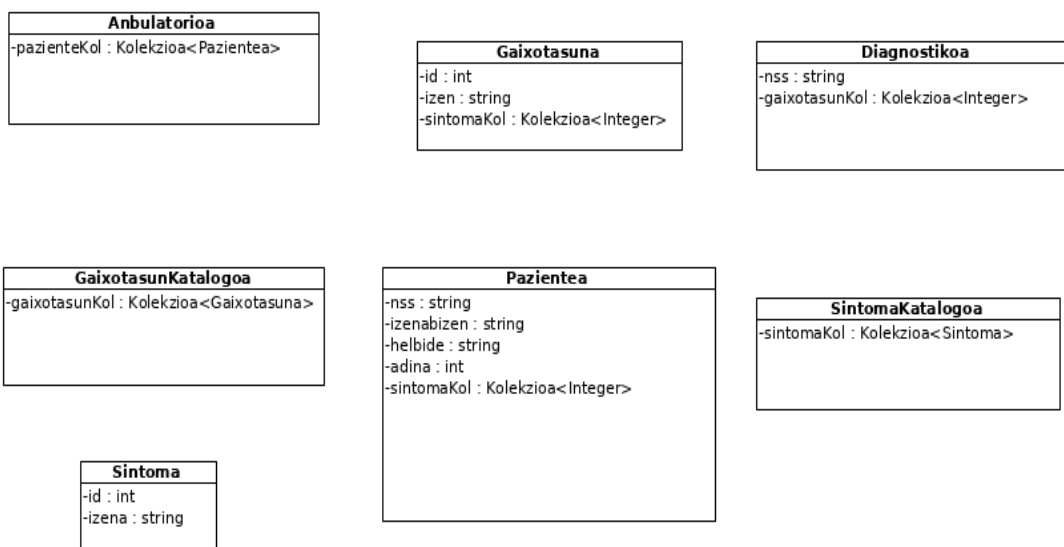
2. E12: Pazienteen diagnostikoa burutu

Paziente guztien diagnostikoak egiteko funtzionalitate bat gehituko dugu, eta, paziente bakoitzari diagnostikoa ere gehituko behar zaio bere ezaugarrien artean. Gure helburua gauzatzeko, osasun-etxeak gaixotasun guztien katalogoa du, eta, gaixotasun bakoitzerako, izena eta sintomen zerrenda gordetzen ditu.

Paziente bakoitzaren diagnostikoa osatzeko, aplikazioak paziente guztien sintomak gaixotasun guztien sintomekin gurutzatuko ditu. Hala, paziente baten diagnostikoan gaixotasun posible bat gehitzeko, azken horren sintomen %75 adierazi behar ditu. Gainera, diagnostiko bakoitzak pazientearen gizarte-segurantzako zenbakia eta bere gaixotasun posible guztiak izango ditu.

Hurrengoa eskatzen da:

- Aurreko *klase diagrama hedatu* aurreko paragrafoan aipatutako funtzionalitatea sartzeko. Ahalik eta aldaketa gutxien sartu. Hemen ere osagai primitiboen erabilera hobetsi ezazu.
- diagSortu():void* metodoari dagokion *sekuentzia diagrama* osatu UML formalismoaren bidez.



Software Ingeniaritza

2018-ko Apirilak 30

IZEN-ABIZENAK: _____

Gremio Koordinazioa

BILBO-ERAIKI enpresak higiezinaren eraikuntza eta berrikuntza obretan beharrezkoak diren gremioak koordinatzen ditu eta kudeatu behar dituen obren zereginen antolaketa automatizatu ahal izateko aplikazio bat diseinatzea eskatu digu.

BILBO-ERAIKI-k obren katalogo bat gordetzen du, non bukatu gabe dauden obra guztiak gordetzen diren. Enpresak obra berri bat onartzen duenean *obra orria* betetzen da. Bertan, bezeroaren izena, obraren kode eta deskribapena eta aurrera eramateko egin beharreko zereginen zerrenda ordenatua adierazten dira. Obraren zeregin bakoitzak bere deskribapena, aurreikusitako iraupena eta gremioaren izena (arotza, igeltseroa, elektrikaria, etab.) izango ditu. Behin beteta, orri hau obra katalogoan gordeko da.

BILBO-ERAIKI-k gainera gremioen katalogo bat dauka, non gremio bakoitzeko bere izena eta lan hori egiten duten enpresen zerrenda bat gordetzen diren. Enpresa bakoitzak lan horren orduko kostea gordetzen du eta agenda bat dauka egin behar dituen zereginekin. Zeregin bakoitzeko zein obretan egin behar den adierazten duen kodea eta zenbatekoa (ordu kopurua*lanaren orduko kostea) gordeko da.

Eskatu diguten aplikazioa gauero 0:00 orduan egikarituko da eta obren katalogoko zereginak bere agendan zeregin gutxien dituen gremio egokiko enpresari esleituko dizkio. Esleipen hau egiteko bere agendan zeregin horren informazioa gordeko da (obren kodea eta lanaren zenbateko totala). Obra baten zeregin guztiak esleituta daudenean obra katalogotik ezabatu beharko da.

Eskatzen da:

- Objektuak identifikatu (Singleton eta DEA), beraien atributuak, konstruktoreak eta problema ebazteko beharrezko operazioak adieraziz. Klase bakoitza UML formalismoa erabiliz adierazi.
- UML diagrama marraztu klase arteko mendekotasunak erakutsiz. Kutxetan klaseen izena bakarrik ipini.
- zereginakEsleitu()* metodoaren sekuentzia diagrama diseinatu, enuntziatuan adierazten duen funtzionalitatea eginez.

Software Ingeniaritza

2019ko Apirilak 17

IZEN-ABIZENAK: _____

Hegazkin kudeaketa

Airelinea komertzialen hegaldi erreserban laguntzen duen erreminta bat garatu nahi da. Aplikazioa plaza libreak dituzten hegaldien informazioa eman eta hala eskatzen duten bezeroen erreserbak egiteko gai izango da.

Erreserba bat egin nahi denean, bezeroak egin nahi duen bidaiaren jatorri eta helmuga hiriak adieraziko ditu eta aplikazioak plaza libreak dituzten hiri horien arteko hegaldiak erakutsiko dizkio, bezeroak gehien komeni zaiona aukeratu dezan. Ondoren, bezeroak hobeto datorkiona aukeratu du aplikazioak erreserba egin dezan. Bezeroaren eskaera betetzen duen hegaldirik ez balego, aplikazioak egoeraren berri emango dion mezu bat pantailaratuko du.

Ez dago ... eta ... hirien artean plaza librerik duen hegaldirik

Hegaldi bakoitzak bere hegaldi zenbakia, hegazkin-konpainia, jatorri eta helmuga hiriak, eguna, aurreikusitako irteera eta iriste orduak, plaza normal baten prezioa eta onartzen dituen plaza kopuru maximoa gordeko ditu. Horretaz gain, dituen erreserba zerrenda ere gordeko du. Bezero baten erreserbak hegaldi zenbakia, bezeroaren identifikatzailea eta bere kreditu txartel zenbakia izango ditu.

Erreserba bat egitean datuak hegaldiaren erreserba zerrendan gorde behar da, ondoren bidai txartela inprimatuko delarik. Txartelak bezeroaren identifikatzailea, hegaldi zenbakia, hegaldia zein konpainiatokoa den, bidaiaren jatorri eta helmuga hiriak, data eta irteera eta iriste orduak eta prezioa izango ditu. Erreserbaren prezioa hegaldian gordetako prezioan oinarrituz kalkulatu da, bezeroak azken urtean konpainia berarekin egindako bidai bakoitzarengatik %2ko deskontua egiten zaiolarik. Hau da:

$$\text{prezioa} = \text{prezio normala} - (0,2 * \text{bezeroak konpainia berarekin azken urtean egindako bidai kopurua})$$

Sortutako bidai txartelak historiko batetan gordeko dira, kudeatutako erreserben estatistika desberdinak kalkulatu ahal izateko. Beste alde batetik, bezeroak edozein momentutan bere erreserba bat ezeztatu dezake, hegaldiko erreserba zerrendatik ezabatuko delarik eta bidai txartela ezeztatuko delarik.

Eskatzen da (erabakiak justifikatuz):

- Objektuak identifikatu (Singleton eta DEA), beraien atributuak, konstruktoreak eta problema ebazteko beharrezko metodoak adieraziz. Klase bakoitza UML formalismoa erabiliz adierazi.
- UML diagrama marraztu klase arteko mendekotasunak erakutsiz. Kutxetan klaseen izena bakarrik ipini.
- hegazkinakBistaratu* (*pJatorri:String*, *pHelmuga:String*) metodoaren sekuentzia diagrama diseinatu. Metodo honek *pJatorri* eta *pHelmuga* hirien artean dauden hegaldiak bistaratuko ditu. *hegaldiaErreserbatu(...)* metodoaren sekuentzia diagrama diseinatu. Metodo honek hegaldi zenbakia, bezeroaren identifikatzailea eta bere kreditu txartel zenbakia emanda aurretik azaldutako erreserba eta bidai txartelaren sorrera egingo du.

Software Ingeniaritza: DISEINUA

2019-ko Maiatzak 24

IZEN-ABIZENAK: _____

OBJEKTUETARA BIDERATUTAKO DISEINUA

Hizkuntza Akademia

Ingeles akademia batek datorren kurtsoko matrikulak automatikoki kudeatzeko aplikazio bat garatu nahi du. Matrikulen kudeaketarako akademiak eskaintzen dituen kurtso guztien zerrenda dugu, kurtsoaren kodearen arabera ordenatuta. Kurtso bakoitzerako kodea, irakaslearen izena, maila, ordutegia, libre dauden plaza kopurua, matrikulatutako ikasleak eta kurtsoarako itzarote-zerrendan dauden ikasleak gordetzen dira.

Ekainaren 1etik irailean klaseak hasi arte, egunero ikasleek plazak libre dituzten kurtsoen batetan matrikula eskatu dezakete. Plaza bat eskatu ahal izateko ikasle bakoitzak bere datu pertsonalak eta aukeratutako hiru kurtsoen kodeak adierazten ditu.

Garatuko den aplikazioak egunaren amaieran jasotako eskaerak prozesatuko ditu dagozkien matrikulak sortuz. Prozesu honetan aplikazioa ikaslea aukeratutako lehenengo kurtsoan matrikulatzen saiatuko da, posible ez bada bigarrenean eta bestela hirugarrenean.

Matrikula kurtsoak plazak libre dituenean bakarrik egin daiteke. Kasu horretan ikaslearen izena matrikulatutako ikasleen zerrendan gehitzen da, plaza libre kopurua gutxitzen da, matrikula sortzen da eta matrikula erregistro orokorrean gehitzen da. Matrikula bakoitzean ikaslearen datu pertsonalak (izen-abizenak, helbidea, telefonoa eta adina), kurtsoaren kodea, irakaslea eta esleitutako ordutegia gordetzen dira. Matrikulak arrakastarik ez badu, ikaslearen izena lehenengo postuan aukeratutako kurtsoan gehitzen da, itzarote-zerrendan.

Eskatzen da:

- a) Klaseak identifikatu (Singleton eta TAD diren adieraziz), beraien atributuak, konstruktoreak eta eskaera kudeaketa egiteko egunero egikaritzen den prozesua aurrera eramateko beharrezko operazioak adieraziz. Klase bakoitza UML formalismoa erabiliz adierazi.
- b) UML diagrama marraztu klase arteko mendekotasunak erakutsiz. Kutxetan klaseen izena bakarrik ipini.
- c) Egunaren bukaeran egikaritzen den matrikulen sorreraren prozesua modelatzen duen sekuentzia diagrama egin, enuntziatuan esandakoa kontutan hartuz.

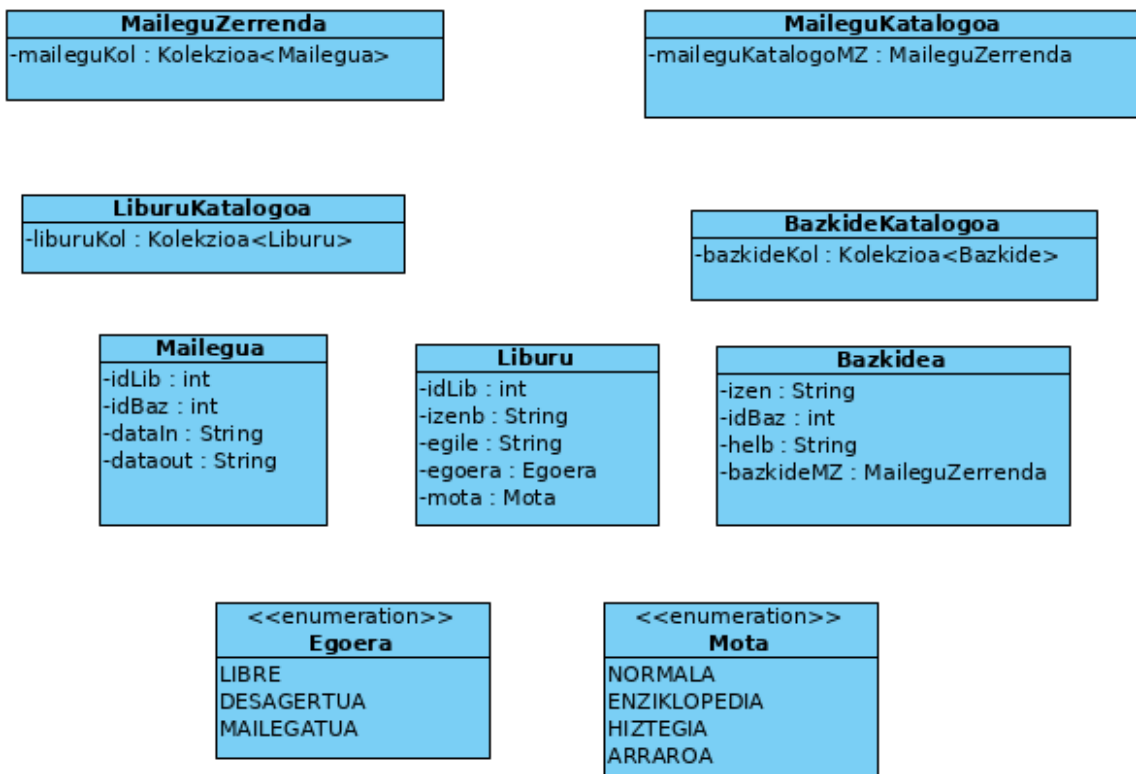
Liburutegia

Liburutegi bat kudeatzeko aplikazio bat inplementatu nahi dugu. Aplikazio horrek ondorengo informazioa gordeko du:

- Bazkide guztien zerrenda: bazkide bakoitzerako izena, kide zenbakia, helbidea eta mailegu guztien zerrenda gordeko ditu. Gehienez, bakide bakoitzak hiru liburu eduki ditzake mailegatuta.
- Liburutegiaren fondoa: liburutegiko ale guztien informazioa gordetzen du. Liburu bakoitzerako, identifikazio kodea, izenburua, idazlea, egoera (libre, maileguan edo desagertua) gordeko ditu. Mailegatu ezin diren liburuaren artean entziklopediak, hiztegiak eta arraroak ditugu.
- Mailegatutako aleen zerrenda: mailegatutako ale bakoitzerako, liburuaren kodea, kidearen zenbakia, maileguaren sarrera-data eta itzultze-data (mailegu denbora maximoa 15 egun dira) gordeko ditu.

Aplikazio honetan liburu baten mailegua burutzeko funtzionalitatea diseinatu nahi dugu, eta hurrengoak eskatzen da:

1. Objetoak identifikatu (Singleton eta TAD), bere atributoak (tipoarekin) eta ariketa ebazteko beharrezkoak diren eragiketak adieraziz. Klase bakoitza UML formalismoa erabiliz adierazi behar da.
2. Klaseen arteko menpekotasunak erakutsi UML diagrama bidez.
3. *liburuMailegatu* funtzionalitatearen sekuentzia diagramak marraztu. Operazio horrek hurrengo egiaztapenak egin beharko ditut: alde batetik, liburu mailegagarria dela, eta, bestetik, bazkidea ez dela mailegu kopuru maximoa ailegatu. Eraitza moduan, liburutegiko informazioa eguneratu behar izango da; hots, kideak mailegatutako aleen zerrenda, liburutegiak mailegatutako aleen zerrenda eta liburuaren egoera liburutegian.



Muntaia-katea

Muntaia-kate bateko makina baten aste bateko (118 ordu) karga egutegia antolatzeko aplikazio bat diseinatu nahi dugu (makinak 24 ordu eguneko mantendu behar da funtzionamenduan). Horretarako, hurrengo informazioa daukagu:

- Makinak burutu beharreko zeregin zerrenda bat, eskaera dataren arabera ordenatuta. Zeregin bakoitzerako, hurrengo informazioa dugu:
 - Zereginaren identifikadorea
 - Eskaera-data
 - Burutze-denbora (orduetan)
 - Prioritatea: 1etik 5erainoko balioa (1 altuena eta 5 txikiena).
 - Materialen zerrenda: zeregin bakoitzeko beharrezko materialak, eta, material bakoitzerako, bere identifikadore eta kopurua gordetzen duena.
- Stock zerrenda. Materialen stock-a gordetzen du (biltegian material bakoitzarako dauden izakinak), eta material bakoitzeko stock-arentzat, materialaren identifikadorea, izakinak eta hornitzailea gordetzen du.

Gauzak horrela, aplikazioa makinaren asteko zereginen egutegian zereginak txertatzen joango da, hurrengo pausuei jarraiki:

- Zereginak prioritatearen arabera ordenatu.
- Zeregin bat Karga zereginen egutegian kargatu ahal izateko, hurrengo baldintzak bete beharko ditu:
 - Zeregin horren beharrezko materialak eta material horren stock-ak konparatuko dira. Behar besteko stock-ik ez badago, zeregina ezingo da zeregin egutegian kargatu.
 - Zeregin horren burutze-denbora asteko lan-kargari batuz gero, ezingo ditu asteko lanordua maximoa gainditu.
- Zeregin bat egutegian kargatu aurretik, bere materialen erreserba egin.
- Zeregina egutegian kargatu
- Zeregina zereginen zerrendatik ezabatu.

Hurrengo eskatzen da (erabakiak justifikatuz) :

- 1) **Objektuak** identifikatu, beren **atributuekin** eta problema ebazteko **operazioekin** batera. Klase bakoitza UML formalismoa erabiliz adierazi.
- 2) **UML diagrama** osatu, klaseen arteko mendekotasunekin batera.
- 3) **zereginakKargatu()** metodoaren **sekuentzi diagrama** marraztu.

Okupazio-panela

Enpresa batek bere langileen agenda kudeatzeko aplikazio bat nahi du. Horretarako, enpresako langile guztien NANA, izena eta urteko agenda gordetzen ditu. Aipatutako agenda horrek langilearen urteko lanegun guztietako okupazioa adierazten du, data, asteko eguna eta egun horretako zereginen zerrenda adieraziz. Gainera, lanegun horietako zereginak hasiera-ordua, amaiera-ordua, parte hartuko duten langileen zerrenda eta deskripzio baten bidez definitzen dira.

Gure aplikazioaren helburua langile-talde baten "Okupazio Panela" sortzea da, guk aukeratutako lanegun-multzo batentzat. Okupazio Panela osatzeko, aplikazioak langile-talde batek (NAN zerrenda) lanegun-multzo batetako (hasiera-data eta amaiera-data jakin baten artekoa) lanordu guztietan daukan okupazioa aztertuko dugu. Enpresatik adierazi digutenez, beren lanegun bakoitzak 8:00-16:00 artekoa lanorduak ditu.

Okupazio Panela tarteez osatuta egongo da, eta tarte batek lanegun batetako lanordu bat adieraziko du, eta data, ordua eta egoeraz (*libre* ala *okupatuta*) definitzen da. Tarte jakin bat libre dagoela kontsideratzeko, aipatutako langile-taldeko inork ere ez du tarte horretan zereginik izan beharko.

Hurrengoa eskatzen da (erabakiak justifikatuz) :

- 1) **Objektuak** identifikatu, beren **atributuekin** eta problema ebazteko **operazioekin** batera. Klase bakoitza UML formalismoa erabiliz adierazi.
- 2) **UML diagrama** osatu, klaseen arteko mendekotasunekin batera.
- 3) **okupazioPanelaSortu** metodoaren **sekuentzi diagrama** marraztu, langile-talde batentzat eta bi daten artean. Hau burutzeko kontuan hartu beharrekoak,
 - Langile-talde eta lanegun-multzo horretarako tarte guztiak sortuko ditugu
 - betiere, taldeko langile guztien okupazioaren arabera.

Emaizta moduan, Okupazio Panelak sarrerako bi data horien arteko tarte guztiak edukiko ditu. Modu horretan, tarte bakoitzak taldeko langile guztiak data eta ordu jakin batean libre dauden ala ez adierazten du.