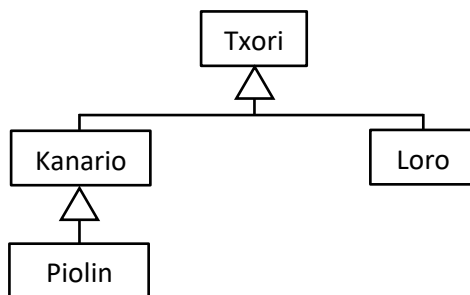


Ariketa01

Implementatu hurrengo hierarkiaren klaseak:



Txori klasearen ezaugarriak:

- Atributu hauek ditu: **sexua** (A [Arra] edo E [Emea], karaktere motakoa), **adina** (zenbaki osoa) eta sortzen diren txorien kopurua zenbatzeko klase atributu bat (estatikoa).
- Bi atributuen datuak jasotzen dituen eraikitzaile bat du. Eraikitzaile honek txorien kopurua ere eguneratuko du.
- **getTxoriKop** izeneko metodoa du. Uneko txori kopurua itzultzen du.
- **norNaiz** izeneko metodoa du. Bere bi atributuen datuak erakusten ditu.

Kanario klasearen ezaugarriak:

- Atributu bezala **sexua**, **adina** eta **tamaina** (zenbaki hamartarra) ditu.
- Kanario klaseak eraikitzaile bakarra du eta hiru atributuen balioak jasotzen ditu.
- **neurtu** izeneko metodoa du. Metodo honek tamainaren arabera mezu bat erakutsiko du: handia tamaina > 30 bada, ertaina tamaina 15 eta 30 artean badago eta txikia tamaina < 15 bada.

Loro klasearen ezaugarriak:

- Atributu bezala **sexua**, **adina**, **zonaldea** (I [Iparraldea], H [Hegoaldea], E [Ekialdea], M [Mendebaldea], karaktere motakoa) eta **kolorea** (karaktere-katea) ditu.
- Eraikitzaile bakar bat du eta atributu guztien balioak jasotzen ditu parametro bezala.
- **nongoaZara** izeneko metodoa du. Metodo honek loroaren zonaldearen arabera Iparraldea, Hegoaldea, Ekialdea edo Mendebaldea hitzak inprimatuko ditu.

Piolin klasearen ezaugarriak:

- Bere atributuak **sexua**, **adina**, **tamaina** eta agertutako pelikula kopurua (zenbaki osoa) izango dira.
- Eraikitzaileak atributu guztien balioak jasoko ditu parametro bezala eta hasieratuko ditu.
- **main** metodoa izango du. Metodo honek ondorengo atazak egingo ditu.
 - Piolin eta loro klaseen instantzia bana sortuko ditu.
 - **norNaiz** metodoa deituko da instantzia bakoitzean.
 - Piolinen instantzian **neurtu** metodoari deituko zaio.
 - Loroaren instantzian **nongoaZara** metodoari deituko zaio.
 - Piolinen instantziaren tamaina aldatuko da eta berriro deituko zaio **neurtu** metodoari.
 - Loroaren instantziaren zonaldea aldatuko da eta **nongoaZara** metodoari deituko zaio.
 - Txori kopurua erakutsiko da.

Ariketa02

Liburutegi batek bi motatako elementuak uzten ditu maileguan: liburuak eta diskoak. Bi kasuetan amankomuna den informazio bat gordetzen da: identifikazio kodea, izenburua eta egilea. Liburuen kasuan aurreko datuez gain orrialde kopurua gordetzen da eta diskoen kasuan diskografiaren izena gordetzen da.

Erabiltzaileak liburutegian mailegu bat egiteko NANA eta izena eman behar dute eta gehienez 5 mailegu egiteko aukera dute. Eskaera bakoitzeko maileguan emandako data eta itzulera data gordetzen da.

Honakoa eskatzen da:

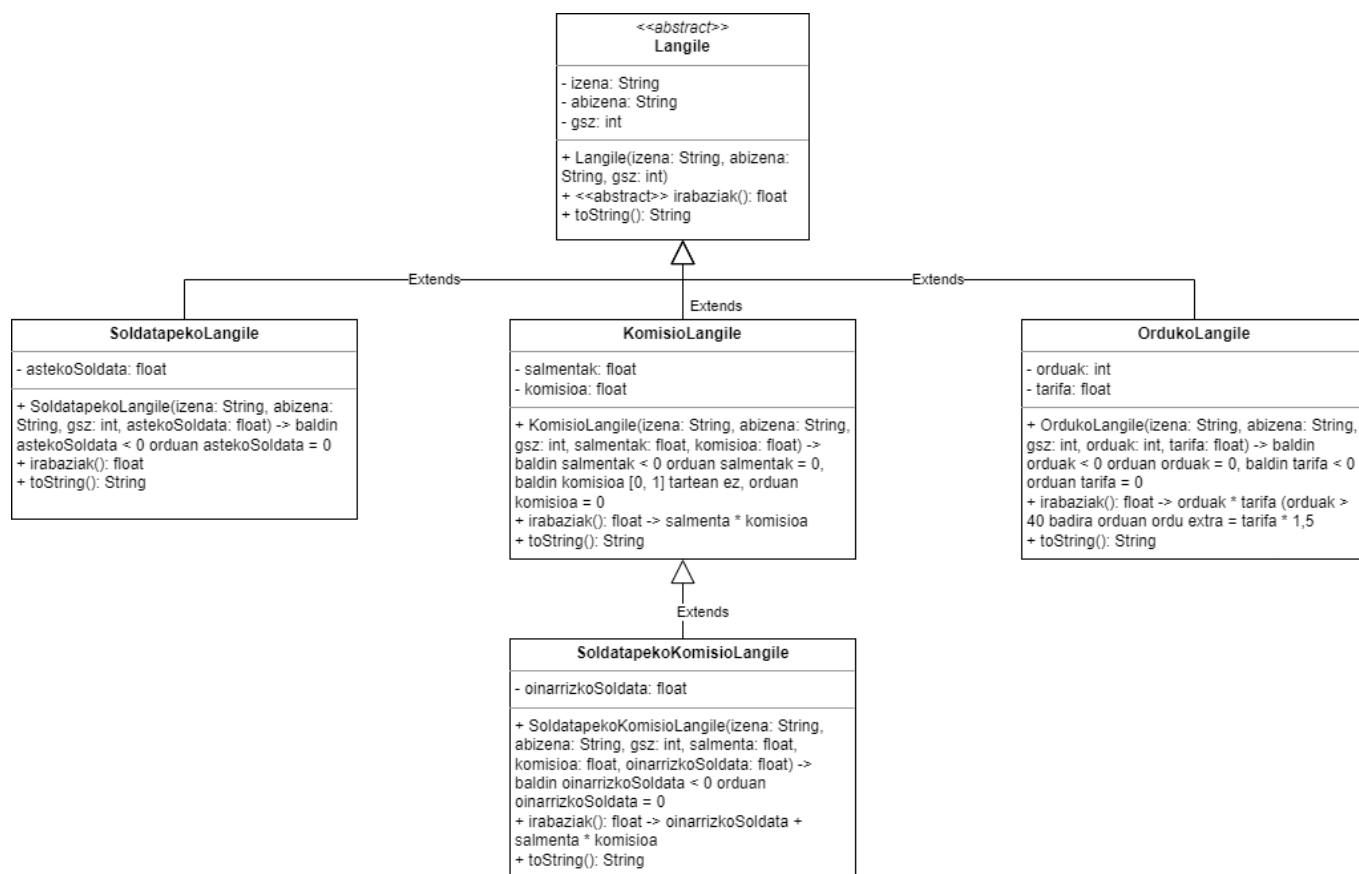
- Sortu **elementuak** paketearen barruan **Elementu** klasea eta honen azpiklase diren **Liburu** eta **Disko** klaseak.
 - Elementu klaseak eraikitzaileaz gain erakutsi metodo bat izango du elementuaren datuak pantailan erakusteko.
 - Liburu klaseak eraikitzailea eta erakutsi eta **kodeOsoa** metodoak izango ditu. Azken metodo honek liburuaren kode, izenburu eta orrialde kopuruarekin osatutako karaktere-kate bat itzuliko du.
 - Disko klaseak eraikitzaile eta erakutsi metodoa izango ditu.
- Sortu **eskaerak** paketearen barruan **Erabiltzaile** eta **Mailegu** klaseak.
 - Bi klaseek eraikitzaile eta erakutsi metodoa izango dute.

Bukatzean egin ondorengo aldaketak eta hausnartu egiten diren galderen inguruan:

- Eman lehenetsitako ikuspen maila elementu guztiei (klase, atributu eta metodo) eta sortu **Main** (*main* metodoarekin) izeneko klase bat beste pakete batean sortutako egitura probatzeko. *Import* agindua erabili behar izan al duzu? Gogoratu: lehenetsitako ikuspen maila atzipen aldagailurik erabili gabe lortzen da.
- Aldatu **Elementu** klaseko atributuen ikuspen maila pribatura. Zer gertatzen da? Probatu berdina babestu eta publiko bezala ezarriz.
- Aldatu **Elementu** klaseko erakutsi metodoaren ikuspen maila. Zer gertatzen da?
- Aldatu **Elementu** klaseko atributuen ikuspen maila pribatura. Gehitu beharrezkoak ikusten dituzun getter eta setterrak **Liburu** klaseko **kodeOsoa** metodoa funtziona dezan.
- Ezarri ikuspen maila minimoa klase eta elementu guztietan definitutako egitura probatzeko.

Ariketa03

Implementatu ondorengo UML klase diagrama eta sortu **Main** izeneko klase bat egitura hauek probatzeko. Sortu langileen klaseak **langileak** paketea eta Main klasea beste pakete batean (lehenetsia nahi bada).



Sortu orain **Ordaingarri** izeneko interfaze bat pakete berdinean. Interfaze honek **kobratzekoFalta** metodo abstraktua izango du eta zenbaki hamartar bat jasoko du parametro bezala eta beste zenbaki hamartar bat itzuliko du. **Langile** eta **KomisioLangile** klaseek ez dute metodo hau implementatzen. Beste klaseek bai implementatzen dute eta irabazien zati bat kobratu dutela adierazteko erabiliko da. Ordaindu zaien zatia hartuko du parametro bezala eta ordaintzeko falta zaiena itzuliko du.

Sortu **Faktura** izeneko klase bat **autonomoak** izeneko beste pakete batean. Klase honek identifikadore bat (karaktere-katea) eta zenbateko bat (zenbaki hamartarra). Klase honek `toString` metodoa berridatziko du eta **Ordaingarri** interfazea inplementatuko du.

Ariketa04

Berreskuratu Ariketa01-eko klaseak eta kopiatu beste proiektu batean. Egin klase hauen gainean ondorengo aldaketak:

- Ezabatu **Piolin** klasean dagoen *main* metodoa.
- Erabili kapsulatzearen printzipioaren jarraibideak. Klase guztietako atributuak pribatuak izango dira. Behar den kasuetan sortu beharrezkoak diren *getter* eta *setter* metodoak eta moldatu metodoen ikuspen maila minimoa izateko.
- Sortu **DatuPertsonal** izeneko klase berri bat. Klase honek bi atributu pribatu izango ditu: **txorilzena** eta **jabelzena**, biak karaktere-kate motakoak. Bi balio hauek jasotzen dituen eraikitzaile bat eta atributuak atzitzeko *getter* eta *setter* metodoak izango ditu.
- Gehitu **Txori** klaseari **izenak** izeneko eta **DatuPertsonal** motako atributu berri bat eta moldatu beharrezkoak diren eraikitzaileak.
- Bihurtu **Txori** klasea abstraktu eta gehitu bertan **abestu** izeneko metodo abstraktu bat. Metodo hau **Piolin** eta **Loro** klaseek implementatuko dute. Agian beste klaseren bat ere abstraktu bihurtu beharko da. Piolinen kasuan honakoa erakutsiko abestu metodoari deitzean: “Nire izena XXX da eta pio-pio abesten dut”. Loroaren kasuan, aldiz, honakoa erakutsiko du: “Nire izena XXX da eta lorito bonito abesten dut”. Egin beharrezkoak iruditzen zaizkizun beste aldaketak.
- Sortu *main* metodoa duen Main izeneko klase bat (pakete berdinean). Metodo honek hurrengoa egingo du:
 - Sortu 3 piolin beraien datu guztiekin eta erakutsi hiru piolinen izenak (berea eta jabearena).
 - Aldatu lehen piolinaren izenak (biak) eta berriro erakutsi datuak.
 - Sortu loro bat bere datu guztiekin.
 - Erakutsi txori kopurua.
 - Sortu 10 tamainako txorien array bat eta gehitu sortutako txoriak bertan.
 - Egin txori guztiek abestea.

Ariketa05

Sortu **enpresa** izeneko pakete bat eta jarraian adierazitako pausoak garatu. Mantendu klase guztietako atributuak pribatu bezala eta sortu *getter* eta *setter* metodoak bakarrik behar direnean.

- **Pertsona** klasea sortu:
 - Atributuak: **izena** eta **NAN** (biak karaktere-kateak) eta **altuera** (zenbaki osoa).
 - Eraikitzaile bakarra du eta honek atributuen hiru balioak jasotzen ditu.
 - **toString** metodoa berridatzi. Metodo honek izena eta altueraz osatutako karaktere-katea itzuliko du.
- **Langile** klasea sortu:
 - Pertsona klasea hedatzen du.
 - **diru** izeneko eta zenbaki oso motako atributu bat du.
 - Eraikitzaile bakarra du eta balio guztiak jasotzen ditu.
 - **diruaIrabazi** eta **diruaGastatu** metodoak izango ditu. Metodo hauek langilearen dirua eguneratuko dute parametro bezala pasatzen zaien kantitatearekin.
 - **toString** metodoa berridatzen du. Izenaz eta altueraz gain duen dirua ere itzuliko du.
- **Mugikor** klasea sortu:
 - Bi atributu ditu: **zenbakia** (karaktere-katea) eta **bateria** (zenbaki osoa, 0 eta 10 tartean).
 - Eraikitzaile bat izango du eta bi balio jasoko ditu parametro bezala.
 - **erabili** izeneko metodo bat izango du. Metodo honek bateria batean txikituko du.
- Moldatu **Langile** klasea **telefonoa** izeneko eta **Mugikor** motako atributu bat izan dezan. Eguneratu ere klasearen eraikitzailea.
- Gehitu **lanEgin** izeneko metodoa **Langile** klasean. Metodo honek langilearen dirua 10 unitatetan handituko du eta mugikorraren bateria erabiliko du.
- Eguneratu **Langilearen toString** metodoa bateriaren kantitatea ere itzuli dezan.
- Sortu *main* metodoa duen **Main** klasea eta honakoak egin:
 - Sortu langile eta mugikor bat eta erakutsi langilearen datuak.
 - Egin langileak bitan lan egitea eta berriro erakutsi bere datuak.
- **Enpresa** klasea sortu:
 - Atributuak: **izena** (karaktere-kate motakoa), **irabaziak** (zenbaki oso motakoa) eta **jabea** eta **gerentea** (biak Langile motakoak).
 - Eraikitzaile bakarra izango du. Honek balio guztiak gerentea ezik jasoko ditu.
 - **kontratatuGerentea** izeneko metodo bat du. Metodo honek **Langile** bat jasotzen du parametro bezala eta gerente atributuari esleitzen dio.
 - **lana** izeneko metodo bat du. Metodo honek enpresako langileak lan egitea egiten du eta enpresaren irabaziak 100 unitatetan handitzen du.
 - **toString** metodoa berridatzen du. Metodo honek enpresaren datu guztiak itzuliko ditu, bere langileen datuak barne.
- Sortu *main* metodoa duen **Main2** izeneko klasea eta honakoak egin:
 - Sortu enpresa bat.
 - Kontratatu gerente bat.
 - Erakutsi enpresaren datuak.
 - Egin enpresak bi aldiz lan egitea.
 - Erakutsi berriro enpresaren datuak.