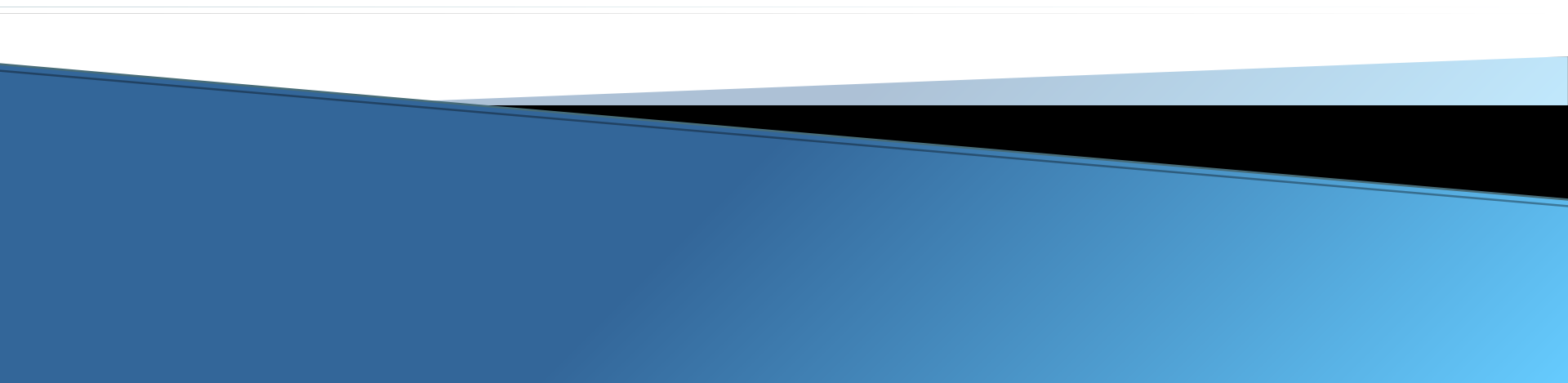
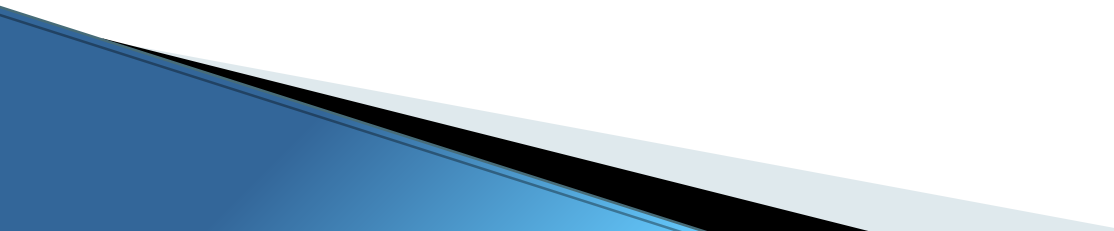


DISEINUAREN OINARRIAK

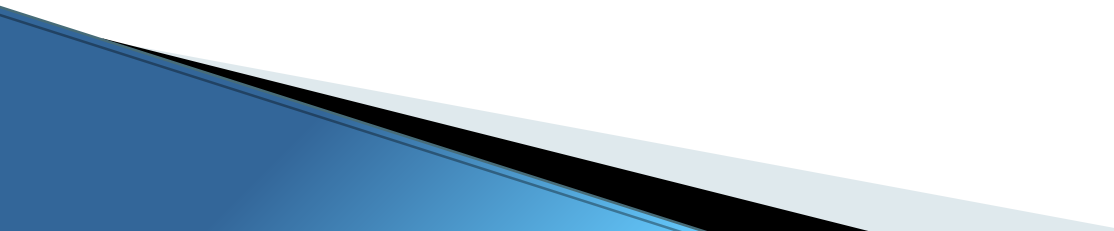
SOFTWARE INGENIARITZA



EDUKIAK

- ▶ Historia apur bat
 - ▶ Softwarearen kalitatea
 - ▶ Diseinuaren oinarriak
 - ▶ Diseinuaren metodologia
 - Diseinu egituratua
 - Objektuei orientatutako diseinua
- 

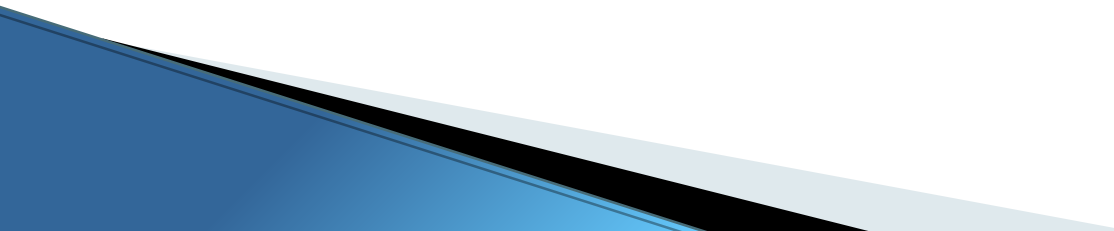
HISTORIA APUR BAT

- ▶ 60. hamarkadaren amaieran hardwarea estandarragoa, prestazio gehiagokoa eta merkeagoa
 - ▶ Softwarearen kalitatea eta efizientzia hobetzeko beharrizana sortu, kosteak merketuz.
- 

HISTORIA APUR BAT

- ▶ Software krisiaren ezaugarriak:
 - Proiektu planifikazio eta kostu estimazio lausoak
 - Softwarea kalitate baxukoa, txarto egituratua
- ▶ Arazoen iturri:
 - Garapena talde txikietan ondo zebilen. Txarto eskalatu zuen talde handiagoetara pasatzean.
 - Aldaketa teknologikoa azkarra
- ▶ **Konponbidea: Software Ingeniaritza!**

HISTORIA APUR BAT


- ▶ SI, orohar, garapen handietarako erabili
 - Lan talde handiek parte hartu
 - Batzuetan, software garapeneko hainbat enpresa
 - ▶ Garapen txikietan ere SI oinarriak aplikatzea gomendagarria
 - Kalitatea bermatu
 - Etorkizuneko aldaketen aurrean sendotasuna
 - Mantentze kosteak gutxitu
- 

SOFTWAREAREN KALITATEA

- ▶ Diseinu metodologiek SW kalitatea hobetu:
 - Barneko ikuspuntutik (garapenaren egileak)
 - Kanpoko ikuspuntutik (erabiltzaileak)

SOFTWAREAREN KALITATEAREN BARNEKO FAKTOREAK

Software ingeniariaren ikuspuntutik,
garapenaren ezaugarriak:

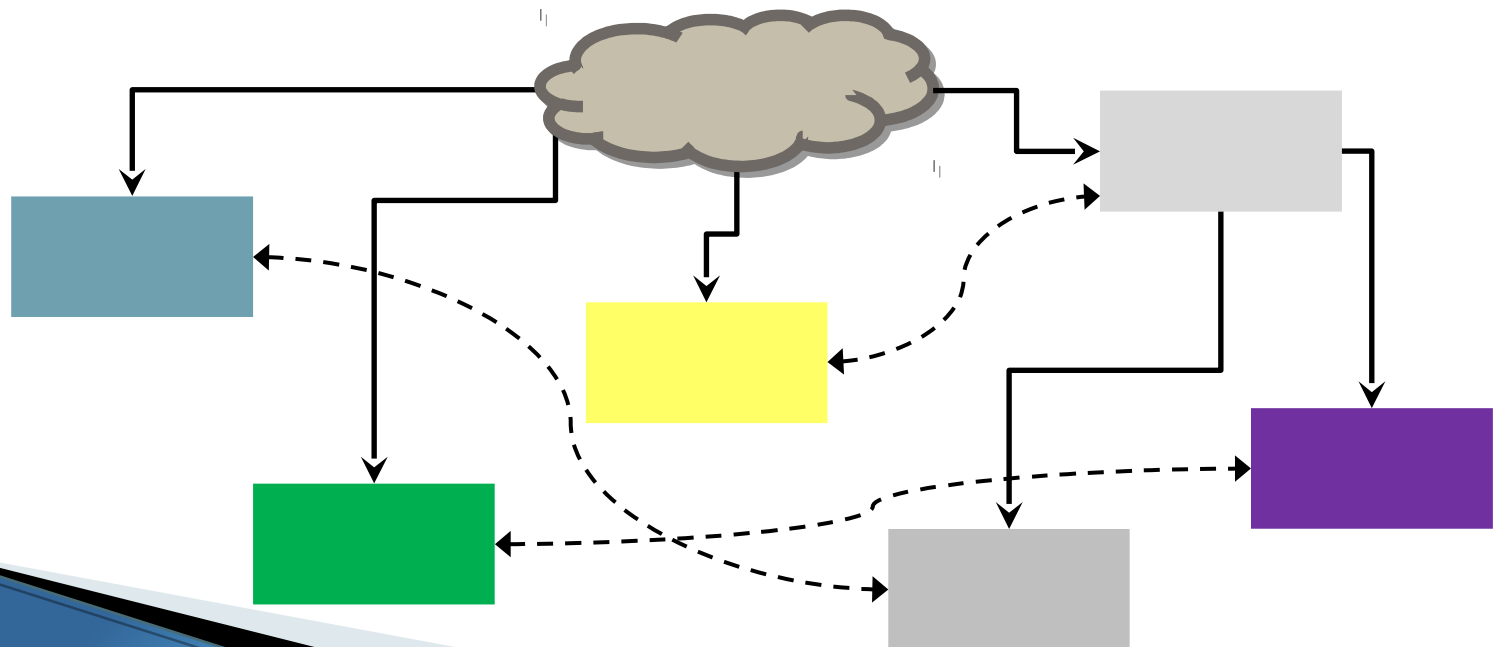
- Ulergarria
 - Irakurterraza
 - Mantengarria
 - Malgua
 - Eramangarria
 - Berrerabilgarria
 - Egiaztagarria
- 

SOFTWAREAREN KALITATEAREN KANPOKO FAKTOREAK

- ▶ **Software erabiltzailearen** ikuspuntutik, garapenaren ezaugarriak ondorengoak izan behar dira:
 - Zuzena
 - Zehatza
 - Erabilterraza
 - Eraginkorra
 - Segurua
 - Sendoa

DISEINUAREN OINARRIAK

- ▶ **Beheranzko diseinu modularra:**
sistema elkar-erlazionatzen diren
moduluetan deskonposatu.



DISEINUAREN OINARRIAK

- ▶ Kontutan hartzeko gaiak:
 - Zein deskonposaketa irizpide erabili?
 - Zer da modulu bat?
 - Nola identifikatu modulu bat?
 - Zelan interkonektatzen dira moduluak?
 - Zelan komunikatzen dira moduluak?

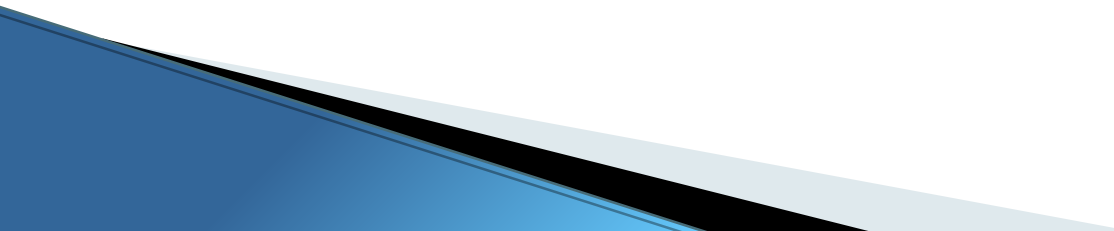
Erantzun ezberdinek diseinu metodologia ezberdinetara eramaten dute.



MODULUEN EZAUGARRI KOMUNAK

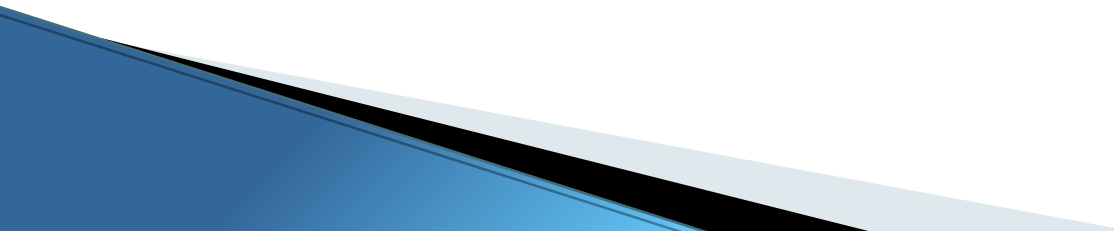
- ▶ Modulu bat “baliabideen” edukiontzia da, eta ondorengo ezaugarriak ditu:
 - Informazioaren ezkutaketa
 - Kohesioa
 - Akoplamendua

INFORMAZIO EZKUTAKETA

- ▶ IDEIA → Modulu bat emanda, ezberdindu :
 - ZER eta NOLA egiten du
 - Erabilera eta funtzionamendua
 - ▶ METODOA → Moduluak bi zatitan banatu:
 - Interfazea
 - Inplementazioa
- 

INFORMAZIO EZKUTAKETA

► Interfazea:

- Zati publikoa, ikusgai dagoena.
 - Erabiltzaileari eskeinitako zerbitzua deskribatu.
 - Erabiltzeko modua adierazi (instrukzioak).
 - Erabiltzaileari zuzenduta.
- 

INFORMAZIO EZKUTAKETA

- ▶ Inplementazioa:
 - Moduluaren zati pribatua, ezkutatua.
 - Eskeinitako zerbitzuen funtzionamendua definitu.
 - Erabiltzailearentzat garrantzitsuak ez diren xehetasunak ezkutatu.
 - Garatzaileak bakarrik ezagutu.

KOHESIOA

- ▶ Ezaugarri intra-modularra.
- ▶ Modulu baten edukien arteko erlazioa neurtu.
- ▶ Maila ezberdinak: sendotik ahulera

Helburua → kohesioa maximizatzea

AKOPLAMENDUA

- ▶ Ezaugarri inter-modularra.
- ▶ Moduluen arteko erlazioa neurtu.
- ▶ Maila ezberdinak: sendoatik ahulera

Helburua → akoplamendua minimizatzea

DISEINU METODOLOGIA

SOFTWARE INGENIARITZA

EDUKIAK

- ▶ Sarrera
- ▶ Diseinu metodologia
 - Objektuen identifikazioa: Booch gida
 - Operazioen identifikazioa: Ellis gida
 - Ikusgarritasun ezarpena
 - Implementazioa

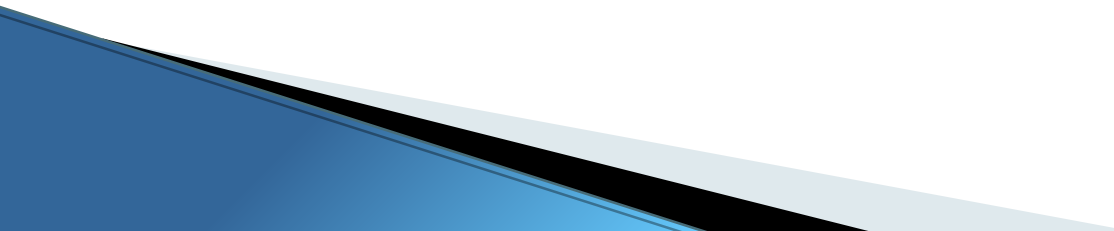
SARRERA

- ▶ *Programazio Lengoaiek* (PL) errealitateko **objektuak** adierazteko mekanismoak hornitu.
- ▶ *Programatzaileak* problemaren objektuen abstrakzioa egin eta inplementu.
- ▶ Objektuak aldatzeko, **algoritmoak erabili**
 - Algoritmoak eragindako irteera, mundu errealeko akzio baten baliokide

SARRERA

Guk definitutako abstrakzioa benetako problematik zenbat eta hurbilago, orduan eta errazago errazago *aldagarritasun, efizientzia, fidagarritasun eta ulergarritasun* helburuetara heltzea

SARRERA

- ▶ **Objektuak eragile** moduan garrantzia, bakoitza bere operazio multzoarekin
 - ▶ Deskonposaketarako irizpide nagusia:
“sistemako modulu bakoitza problemaren objektu edo klase bati dagokio”
 - ▶ **Abstrakzioak/informazioa ezkutatzea**, objektuetara bideratutako garapenaren funtsa
- 

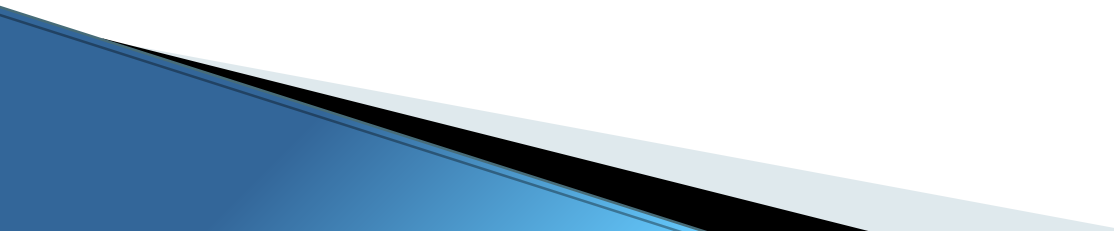
SARRERA

- ▶ **Objektua**

- **Klase** baten **instantzia**
- “Egoera eta jokaera *duen entitatea*”
 - Egoera → atributuak
 - **Jokaera** → operazioak

- ▶ **Programa** → elkarrekin lan egiten duten objektu multzoa
- 

OBJEKTUEI BIDERATUTAKO DISEINU METODOA

1. Objektu klaseak eta beren atributoak identifikatu
 2. Objektu klase bakoitzaren operazioak identifikatu
 3. Objektu klaseen ikusgarritasuna ezarri
 4. Identifikatutako klaseen atributuen ikusgarritasuna ezarri
 5. Klase bakoitzaren implementazioa
- 

OBJEKTU KLASEEN IDENTIFIKAZIOA

- ▶ Problemaren baitako objektuak identifikatu
- ▶ Objektuen atributuak identifikatu
- ▶ Identifikaziorako gidak
 - Booch
 - Ellis

Ikuspegi biak batera egin daitezke



Adibidea: Anbulategia

Durangoko anbulategia bere kontsulten datuen infomatizazioa egiten dabil. Horretarako, anbulategiak gaixoen hurrengo informazioa eman behar du: izena, abizenak, adina, SS zenbakia, probintzia eta sintomen zerrenda. Hala, sintoma bat bere izenarekin eta kodearekin definitzen da.

HELBURUA: sintoma jakinik, sintoma hori daukaten gaixo kopurua bueltatu.

BOOCH GIDA

- ▶ Deskribapeneko **elementuen izenak** kontutan hartu.
- ▶ Izen mota ezberdinak:
 - Izen **arruntak**: entitateak izendatzeko, ad.:mahaia, terminala, sentsorea (klaseak edo DMA-ak)
 - Izen **bereziak** edo **erreferentzia zuzenerako**, ad.: berotasun sentsorea, nire mahaia, enpresa bateko langileen zerrenda,... (EMA-k edo Singleton-ak)

Adibidea (Booch gida)

Durangoko **anbulategia** bere kontsulten datuen infomatizazioa egiten dabil. Horretarako, anbulategiak **gaixo** bakoitzerako hurrengo informazioa eman behar du: izena, abizenak, adina, SS zenbakia, probintzia eta sintomen zerrenda. Hala, **sintoma** bat bere izenarekin eta kodearekin definitzen da.

HELBURUA: sintoma jakinik, sintoma hori daukaten gaixo kopurua bueltatu.

Adibidea (Booch gida)

- ▶ Problemaren deskribapenean agertzen diren izenak:
 - EMA: *Anbulategia*
 - DMA: Gaixoa, Sintoma
- ▶ Klaseak eta atributuak definitu

Adibidea (Booch gida)

Anbulategi (EMA)

- GaixoKol: Kolekzioa

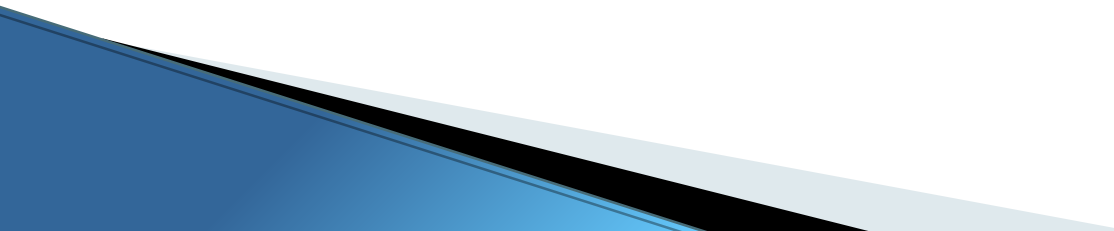
Sintoma

- Izena : string
- Kodea : int

Gaixoa

- izena : string
- abizena : string
- SSzenb : string
- adina : int
- probintzia : string
- sintomaKol: Kolekzioa<Sintoma>

ELLIS GIDA

- ▶ Datu abstrakzioak argitu baino lehen, algoritmo nagusia eskematizatzen du.
 - ▶ Operazioak noiz identifikatu? Aplikazioak burutu beharreko prozesua kontuan hartzerakoan
 - ▶ Objektuak prozesu hori burutzen duten agenteak dira.
- 

Adibidea (Ellis gida)

Durangoko anbulategia bere kontsulten datuen infomatizazioa egiten dabil. Horretarako, anbulategiak **gaixo** bakoitzerako hurrengo informazioa eman behar du: izena, abizenak, adina, SS zenbakia, probintzia eta sintomen zerrenda. Hala, **sintoma** bat bere izenarekin eta kodearekin definitzen da.

Adibidea (Ellis gida)

- **Anbulategia (abiapuntua)**

gaixoKol guztia zeharkatu

- **Gaixoak**

sintomaKol guztian *pSintoma* bilatu

- **Sintoma**

bilatutako sintoma bada, true bueltatu



Adibidea (Ellis gida)

Anbulategia (EMA)

- + sintomaKopLortu(pSintoma:String) : int
- getIterator() : Iterator<Gaixoa>

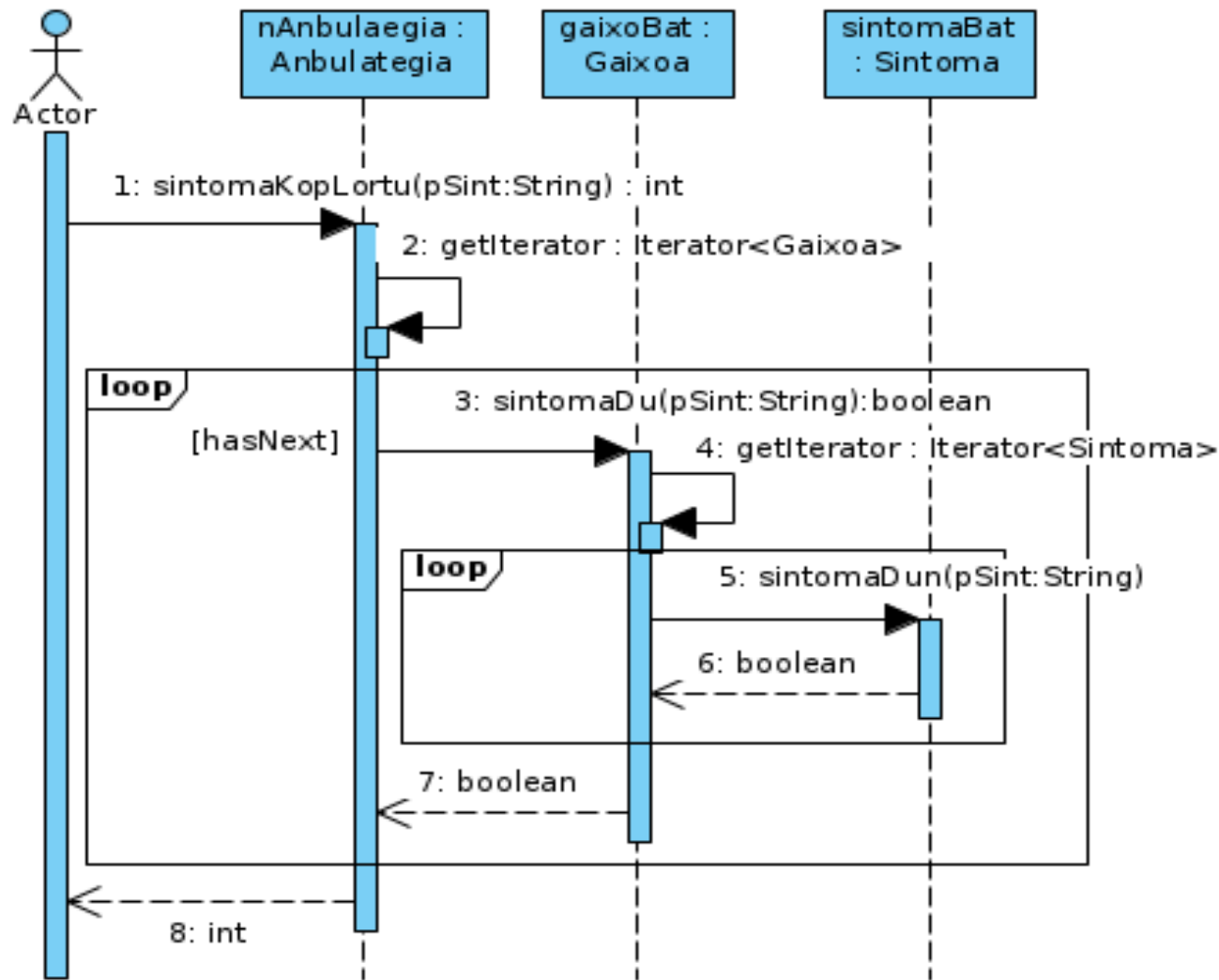
Gaixoa

- + SintomaDu(pSintoma:String) : boolean
- getIterator() : Iterator<Sintoma>

Sintoma

- + SintomaDu(pSintoma:String) : boolean

Sekuentzia diagrama



OPERAZIOAK IDENTIFIKATU

- ▶ Klase batek metodo motak posibleak:
 - **Konstruktoreak**: klasearen instantziak sortu
 - **Sarrera metodoak**: atributuen balioak itzuli
 - **Aldaketa metodoak**: atributuen balioa aldatu.
 - **Produktoreak**: existitzen diren instantzietatik, instantzia berriak sortzen dituzte
 - **Bistaratzeko metodoak**: atributuen balioak erakutsi

IKUSGARITASUNA

- ▶ Klase arteko menpekotasunak identifikatu:
 - Klase batek besteetatik ikusten duena
 - Klase batek besteentzat ikusgarri duena
- ▶ Klaseen **atributuak** beti **pribatuak**.
- ▶ Klaseen **metodoak pribatuak**, defektuz.
 - Klase baten metodo bat beste klase batentzako **ezinbestekoa** bada, bere ikusgarritasuna **publikora aldatu**.

OBJEKTUEN INPLEMENTAZIOA

- ▶ Xehetasunen inplementazioa azkenerako; sistemaren arkitektura eta jokaerei eman pisua.
- ▶ Objektuen adierazpen eta inplementazio ezberdinak saiatu daitezke.

Supermerkatua

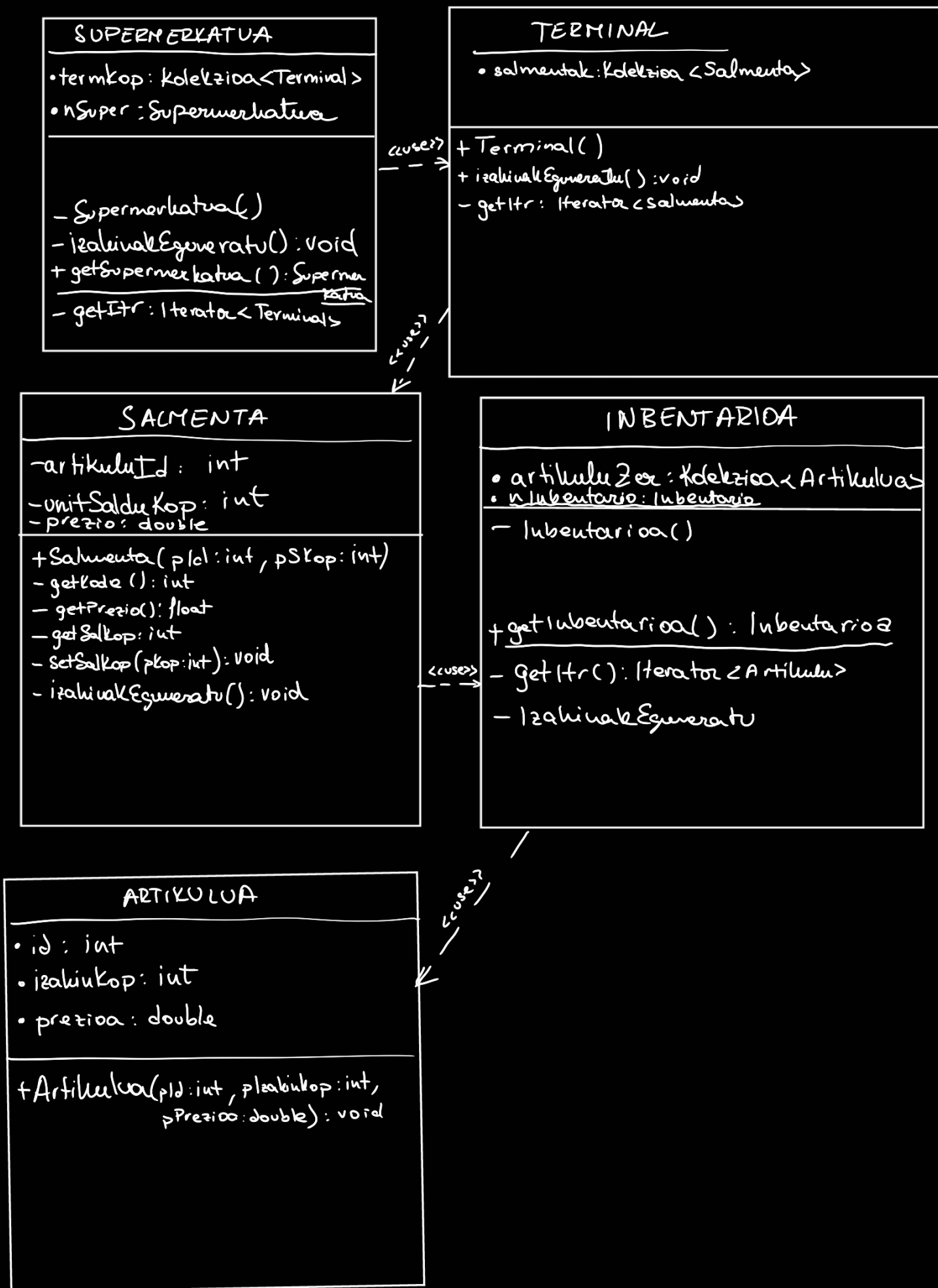
Supermerkatu baten artikuluen izakinen eguneraketa-prozesua garatu nahi dugu. Supermerkatuak 40 terminal ditu kutxa-erregistratzaile moduan, eta, **terminal** bakoitzak egunean zehar egindako salmenta guztien informazioa erregistratzen du. **Salmenta** bakoitzerako, artikuluko kodea eta saldutako unitate kopurua gordeko da.

Eguna bukatzerakoan inbentarioa egingo da, eta azken horretan saldutako artikuluen izakinak eguneratuko dira, 40 terminalerako salementen informaziotik abiatuta. **Inbentarioak** hurrengoak gordetzen ditu artikuluko bakoitzerako: artikuluko kodea, izakin kopurua eta prezioa unitateko.

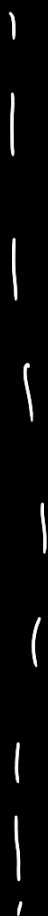
Hurrengoak eskatzen dira:

- a) • Klaseak identifikatu (EMA eta DMA), beren atributu eta problema ebazteko beharrezko metodoekin, UML formalismoa erabilita.
- b) • Egunaren bukaeran exekutatu den *izakinakEguneratu* metodoaren sekuentzien diagrama egin.

a) EMA: Supermerkatura, Inbentarioa
DMA: Terminal, Salmenta, Artikulua



b) $\frac{1}{2} \times \frac{1}{2}$



Liburutegia

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

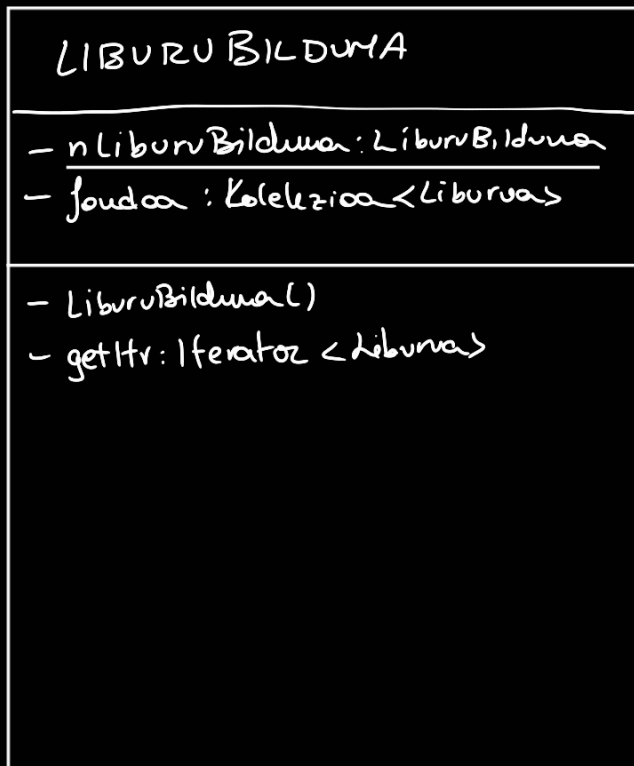
- **Bazkide** guztien zerrenda: bazkide bakoitzerako izena, kide zenbakia, helbidea eta mailegu guztien zerrenda gordeko ditu. Gehienez, bazkide bakoitzak hiru liburu eduki ditzake mailegatuta.
- Liburutegiaren **fondoa**: liburutegiko ale guztien informazioa gordetzen du. Liburu bakoitzerako, identifikazio kodea, izenburua, idazlea, egoera (libre, maileguan edo desagertua) gordeko ditu. Mailegatu ezin diren liburuen artean entziklopediak, hiztegiak eta arraroak ditugu.
- Mailegatutako aleen zerrenda: mailegatutako ale bakoitzerako, **liburu**aren 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 hurrengo eskatzen da:

1. Objetuak 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. *liburuaMailegatu* funtzionalitatearen sekuentzia diagrama marraztu. Operazio horrek hurrengo egiaztapenak egin beharko ditu: alde batetik, liburu mailegagarria dela, eta, bestetik, bazkidea ez dela mailegu kopuru maximora ailegatu. Emaitza moduan, liburutegiko informazioa eguneratuko da; hots, kideak mailegatutako aleen zerrenda, liburutegiak mailegatutako aleen zerrenda eta liburuaren egoera liburutegian.

a) EMA: Liburu_bilduma

DMA: Liburua



<<use>>

— — — — —>

