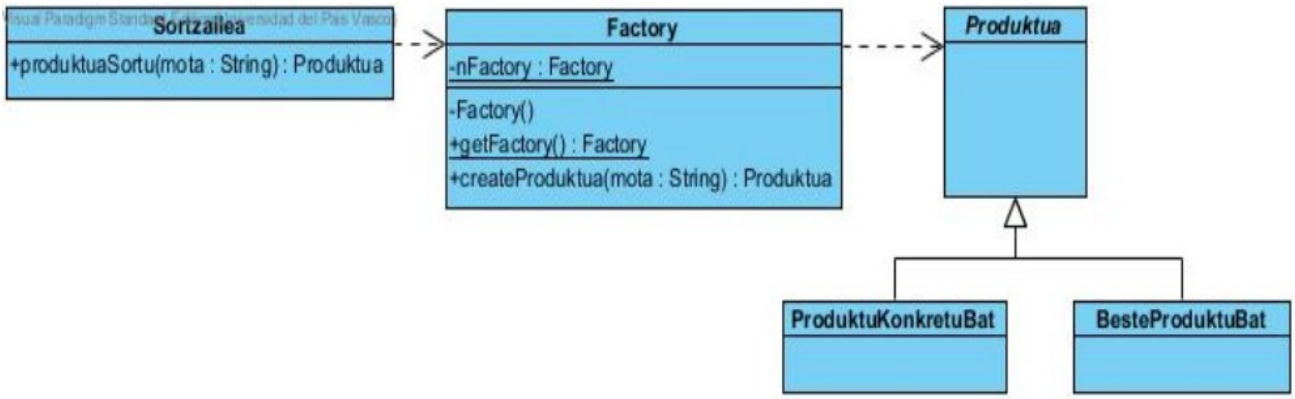
SORTZAILEAK

Factory: objektuak sortzeko interfazea definitu, baina, azpiklaseen esku klaseen instantziazioaren kudeaketa.

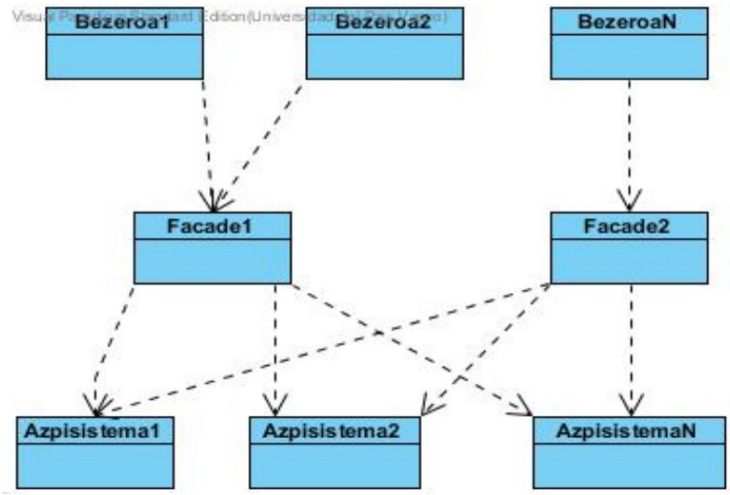


Ezaugarriak

- Objektuen sorrera faktorian enkapsulatu
- Objektuen sorrera (faktoria) eta objektuekin kudeaketa (pizzeria) banatu
- Pizza mota berria sortzeko
  - Klase abstraktua hedatzeko klasea sortu
  - Faktorian bi lerro gehitu
- Objektuen sorrera kontrolatu
- Mantenketa eta hedatzea erraztu

EGITURAZKOAK

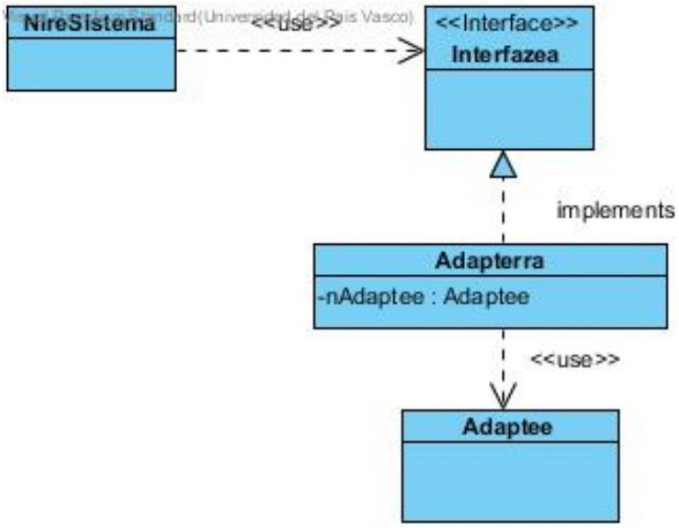
Facade: azpisistema bateko interfaze-multzo bati interfaze bateratua eman; hots, maila altuko intefazea, azpisistema erabilterrazagoa izan dadin.



Ezaugarriak

- Bezeroak azpisistematik isolatu
- Bezero/azpisistema akoplamendu ahula.
- Azpisistemak bezeroarentzat eskuragarri, hala nahi izanez gero
- Sistema geruzatan banatu
- Kontuan izan:
  - Azpisistemen aldaketek Facade aldatu
  - Bezeroek azpisistema desberdinak erabiliz gero, facade desberdinak

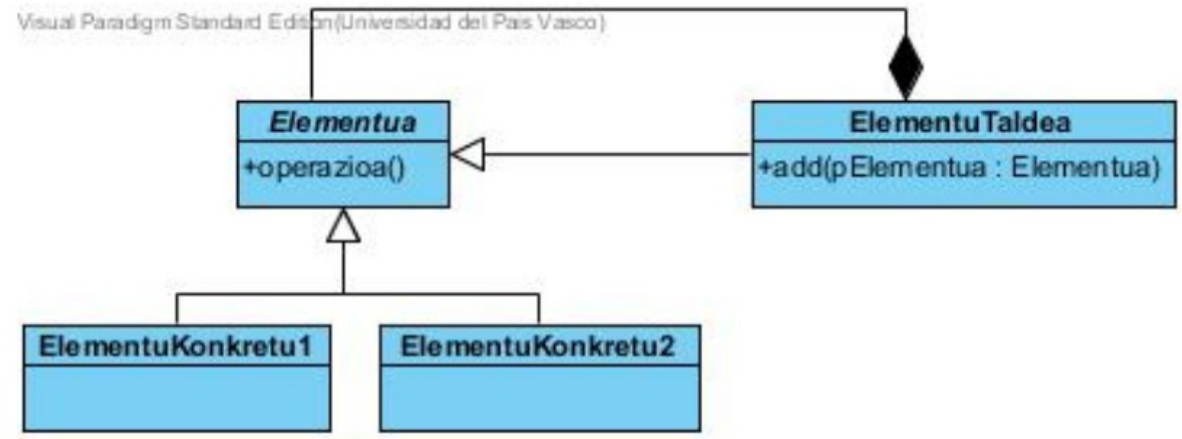
Adapter: klase baten interfaze bezeroak esperotako interfazera bihurtzen du; hots, interfaze bateraezinei elkarrekin lan egiteko aukera eman.



Ezaugarriak

- Berrerabilgarritasuna hobetu
- Hedapena erraztu
- Adapterrak interfazea enkapsulatu
  - Bezeroa interfazetik desakoplatu
  - Interfazea aldatuta, bezeroak ez du ikusten
- Adaptee-a ez da ukitzen

Composite: “part-whole” hierarkiak errepresentatze aldera, objektuak zuhaitz egituretan osatzea ahalbidetu. Bezeroei banako objektuak eta konposatuak uniformeki erabiltzen utzi.



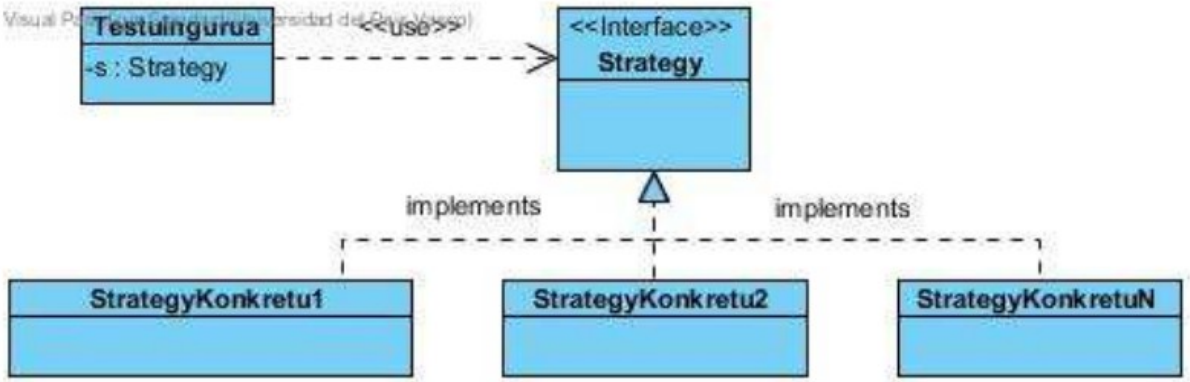
Ezaugarriak

- Banakako elementuak (hostoak) eta konposatuak (nodoak) zuhaitz egitura berean txertatu.
  - “Part-whole” hierarkiak
- Objektu guztiek interfaze bera
  - Nodo zein hosto, era berean tratatu.

PORTAERAZKOAK

Strategy: algoritmo familia bat definitzen du, horietako bakoitza enkaptsulatuz, eta beren artean trukagarri egiten ditu. Algorimoa aldatzea ahalbidetzen du,

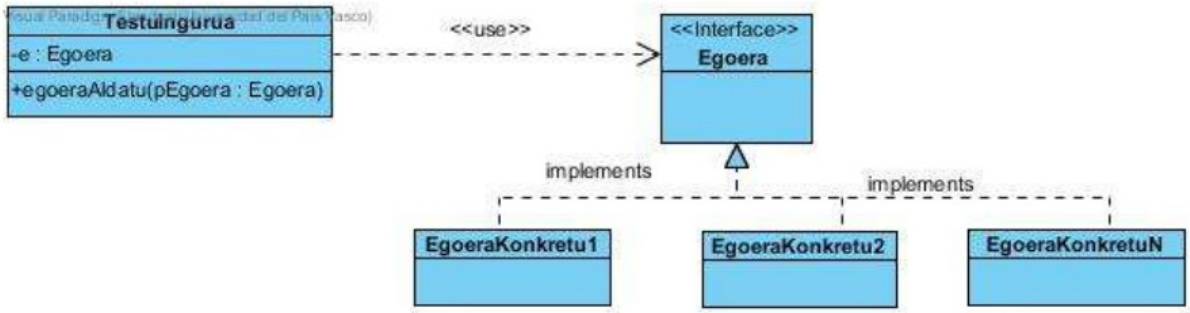
bezeroarekiko independenteki.



Ezaugarriak

- Testuingurua eta portaerak banatu
- Berrerabilpena hobetu
- Algoritmo familiak definitu
- Inplementazio aukera desberdinak
- Bezeroak Strategy desberdinak ezagutu eta aukeratzen ditu

State: objektu baten barne egoera aldatzean, bere portaera aldatzea ahalbidetzen dio.



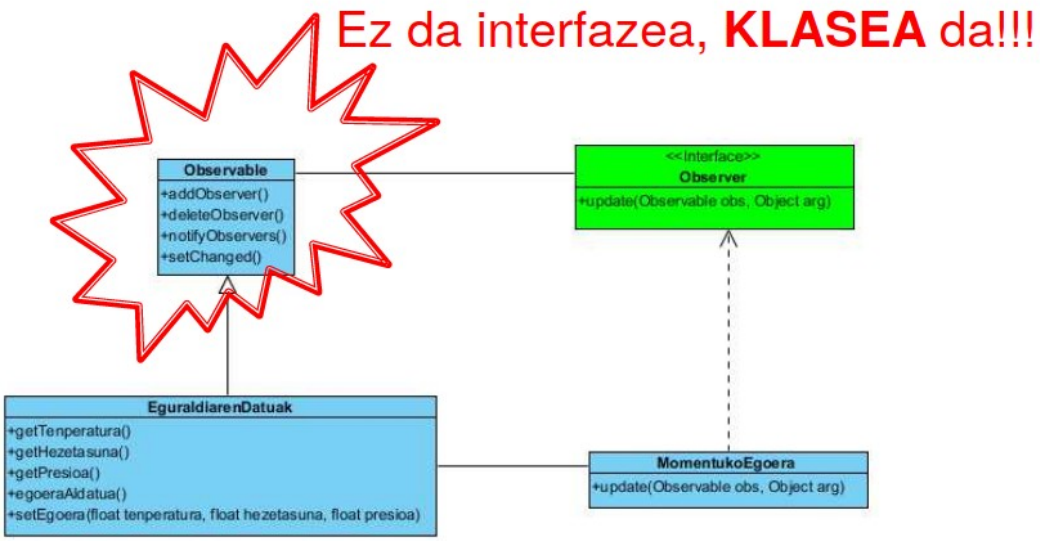
Ezaugarriak

- Egoeraren arabera, portaera desberdina
- Egoera bakoitzeko portaera enkapsulatu
- Egoeren arteko transizioak esplizituak
- Hedagarria
- Bezeroak egoeren inguruko informazio gutxi edo ezer

State/Strategy arteko desberdintasuna: ASMOA!!

- State: portaera multzoak egoera objektuetan kapsulatzea. Bezeroak ez daki ia ezer egoeren inguruan.
- Strategy: “runtime”ean strategy objektua aldatzea ahalbidetu. Bezeroak aukeratzen du zer egin.

Observer: objektuen arteko ”one-to-many” dependentziak definitu; objetu batek bere egoera aldatzen duenean, bere menpeko guztiei jakinzaraziko die.



Ezaugarriak

- Akoplamendu soltea (loose-coupling); *observable*-k ez daki *observer*-aren mota.
- Entzuleei jakinarazpenak bidali
- Hedagarria

Model-view-controller: aplikazio bateko datuak, erabiltzaile-interfazea eta kontrol logika banatu. Hiruretako edozein aldatuta ere, besteen funtzionamenduari eraginik ez.

EGIN KLASE DIAGRAMA!!

Ezaugarriak

- Osagai bakoitza independenteki garatu
- Aldagarritasuna
- Bista anitzak izateko aukera
- Bistek ereduaren zatiak ikusi
- Edozein aplikazio motatara aplikagarriak