DISEINU PATROIAK

DISEINU PATROIAK

MOTIBAZIOA

- OZ diseinua, zaila!! Berrerabilgarritasuna, are zailago!!
- Iraganean funtzionatu duten soluzioak berrerabili
- PATROIEK diseinu problema zehatzak ebatzi
  - Diseinu malgua eta berrerabilgarria ahalbidetu
- SOLID printzipioetan oinarritu

DISEINU PATROIAK

DEFINIZIOA: diseinu problema orokor bat testuinguru jakin batetan ebazteko soluzioa. Azken hori aurretiaz defininitutako objektu eta klaseen konfigurazioa da.

DISEINU PATROIAK

HISTORIA

- 1964-1979: Christopher Alexanderrek patroiak planteatu *arkitektura munduan*.
- 1990-1992: “Gang of Four”(GOF) taldearen lana hasi, informatikara Alexanderren ideiak.
- 1995: ***“Design Patterns, Elements of Reusable Object-Oriented Software”*** liburu ospetsua argitaratu

DISEINU PATROIAK

SAILKAPENA

- **Sortzaileak**: objektuen sorkuntza
- **Egiturazkoak**: klase eta objektuen konposaketa
- **Portaerazkoak:** klase eta objektuen elkarreragin eta ardurak banatzeko era

PATROIAK: ITERATOR

MOTIBAZIOA

- Inplementazio bakoitzerako, osagaiak zeharkatzeko era desberdina.

Array-ak	Zerrenda estekatuak

…
String datua = null;
for (int i= 0; i< zerren.size();i++) {
  datua = zerren.get(i);
  …
}
…
	…
	String datua = null;
	Nodoa aux = zerren.getFirst();
	while (aux!=null) {
	  datua = aux.getContent();
	  …
	  aux = aux.getNext();
	}
	…

PATROIAK: ITERATOR

ARAZOA

- Gauza bera egiteko, inplementazio desberdinak

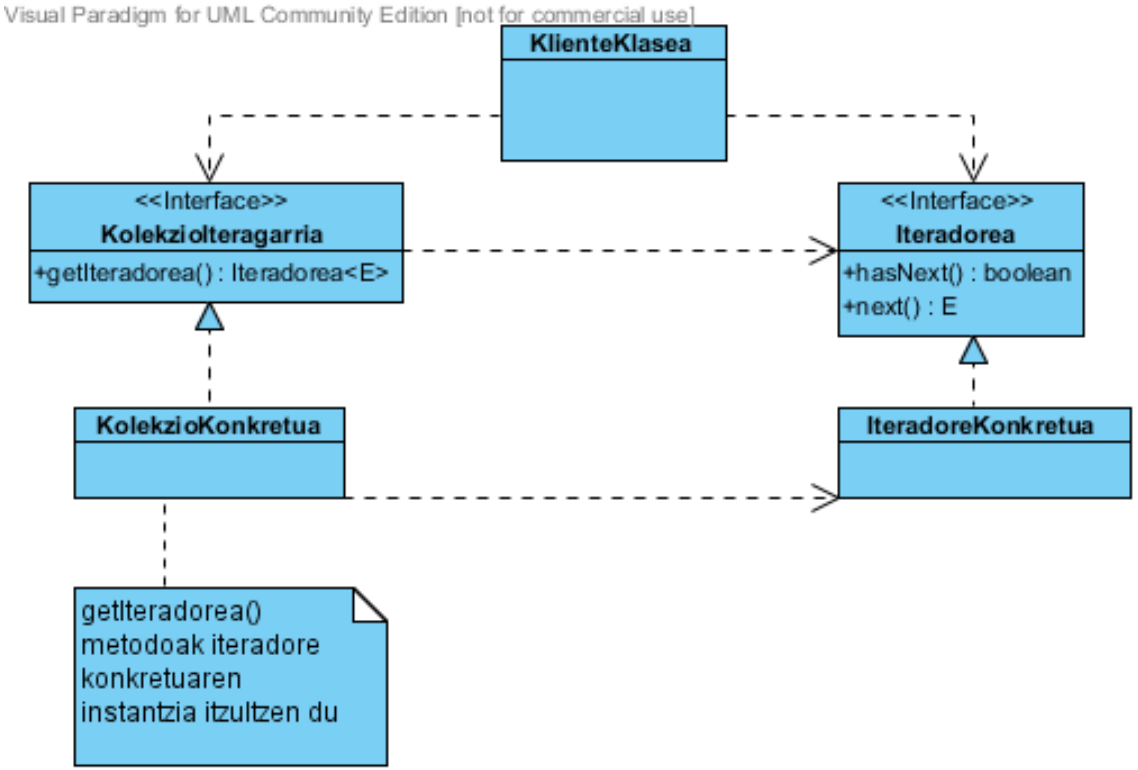
HELBURUA

- Kolekzio bat sekuentzialki zeharkatzeko modu bat lortu, baina, bere adierazpenaren modu independentean

PATROIAK: ITERATOR

DESKRIBAPENA

- Java interfazetan oinarritzen da:



PATROIAK: ITERATOR

INPLEMENTAZIOA

Array-ak

```
ArrayList<String> zerr =
                new ArrayList<String>();
…
String kate = null;
Iterator<String> iter =
                      zerr.iterator();
while (iter.hasNext()){
    kate = iter.next();
    …
}
…
```

Zerrenda estekatuak

```
LinkedList<String> zerr =
    new LinkedList<String>();
…
String kate = null;
Iterator<String> iter =
                      zerr.iterator();
while (iter.hasNext()){
    kate = iter.next();
    …
}
…
```

Ez da kolekzio konkretuaren egitura ezagutu behar osagaiak zeharkatu ahal izateko

PATROIAK: SINGLETON

MOTIBAZIOA

- Elementu bakarra puntu desberdinetatik erreferentziatzeko
  - Beti berdina dela ziurtatu
  - Instantzia bakarra egon behar da, denek objektu bera ikusteko.

PATROIAK: SINGLETON

DESKRIBAPENA

- Klaseak berak sortu bere instantzia bakarra.
- Sarrera globala klase-metodo batekin (**`static`**).
- Klaseko konstruktorea pribatua, instantzia berriak egitea ezinezkoa izateko.

PATROIAK: SINGLETON

INPLEMENTAZIOA

```
public class NireSingleton{
        private static NireSingleton instantziaBakarra = null;
        // Konstruktore pribatua
        private Singleton() {}
        //Instantzia bakarra itzuli.
        //Sortuta ez badago, sortu egiten du
        public static Singleton getInstance() {
            if (instantziaBakarra == null) {
                instantziaBakarra = new Singleton();
            }
            return instantziaBakarra;
        }
    }
```