

SOLID LABORATEGIA

Ireki-ltxia Oinarria (OCP).

1. Covid19Pacient klaseak OCP printzipioa betetzen du?. Hau da, zer gertatuko litzateke beste sintoma mota desberdin bat agertuko balitz (adb: DigestiveSymptom)?. Aplikazio guztia aldatu, OCP printzipioa betetzeko.

Hasiera batean ez du OCP printzipioa betetzen. “Software entitateak irekiak izan behar dira luzapenak onartzeko, baina itxiak kodea aldatzeko”, gure kasuan kodea aldatu beharko genuke luzapen bat gehitzean. Hori dela eta. Symptom klase abstraktua sortu dut eta metodoei hori bera sartuko zaie parametro gisa(ez du jakingo zein metodo den).

2. Aplikazio bat sortu 3 sintoma mota desberdin duen Paziente batekin, eta programa nagusi batean, calcCovid19Impact () metodoari deitzen dion programa bat sortu.

```
public static void main(String[] args) {
    Covid19Pacient hiru_sintomako_pazientea= new
Covid19Pacient("Julen_HiruSintoma", 21);

    int cardiolImpact = 2; int cardiolIndex=1; int cardioAffecteddays=3;
    CardioVascularSymptom cardiovascular= new
CardioVascularSymptom(cardiolImpact,cardiolIndex,cardioAffecteddays);

    int neuroImpact = 3; int neuroIndex=2; int neuroAffecteddays=3;
    NeuroMuscularSymptom neuromuscular= new
NeuroMuscularSymptom(neuroImpact, neuroIndex,neuroAffecteddays);

    int respirImpact = 1; int respirIndex=1; int respirAffecteddays=3;
    RespiratorySymptom respiratory = new
RespiratorySymptom(respirImpact,respirIndex,respirAffecteddays);

    hiru_sintomako_pazientea.addSymptom(cardiovascular,90);
    hiru_sintomako_pazientea.addSymptom(neuromuscular,90 );
    hiru_sintomako_pazientea.addSymptom(respiratory,90 );

    double covid_impact=hiru_sintomako_pazientea.calcCovid19Impact();
    System.out.println(covid_impact);
}
```

3. Egin duzun aldaketa errefaktORIZAZIO bat dela kontsideratzen duzu? Justifikatu erantzuna.

Bai, errefaktORIZAZIOA bat dela kontsidera dezakegu. Kodea errestrukturatu dugu sintomen egitura aldatu dugun momentutik, bere barne-egitura aldatuz baina kanpo-egitura portaera mantenduz.

4. Sintoma bakoitzak *affectedDays* atributu bat edukiko balu, sintoma hori irauten dituen egunekin, eta *Covid19Pacient* klaseak *sanatedDays()* metodoa bat gehitu nahiko bagenu, pertsona noiz sendatuko den itzultzen badu (metodoak, egun gehien irauten duen sintoma itzultzen du). Nola osatuko zenuke klaseak? Programa nagusia osatu, eskakizun hau frogatzeko.

```
public double sanatedDays() {
    double maximoa = Maximoa_aurkitu();
    return maximoa;
}

private double Maximoa_aurkitu() {
    double maximoa=0;

    for (Symptom c:symptom_hashmap.keySet()) {
        int afektatutako_egunak=c.getAffectedDays();
        if(afektatutako_egunak>maximoa) {
            maximoa=afektatutako_egunak;
        }
    }
    return maximoa;
}
```

Kode hontaz gain, sintomen klaseetan aldaketak egin ditut sintoma berriek *Symptom* klasea hedatuaraziz.

Erresponsabilitate bakarreko printzipioa (SRP)

Galderak:

1- 2. Aplikazioa errefaktoriazatu, erresponsabilitate bakoitza klase isolatu batean geratu dadin. Adierazi zein aldaketa egin beharko litzateke afekzioaren balioa (hau da, *afection*) kalkulatzeko, *covidImpact*>50 handiagoko sintomak soilik kontuan hartzen badira.:

Lehenik *Afection* eta *Increment* klaseak sortu ditut. Ondoren, *Afection* klasean *AfectionKalkulatu()* metodoa gehitu dut sintomen hashmap-a duelarik eta *Afection* klasearen metodoan hashmap-a korritzean *covidImpact*>50 delakoa konprobatuko du. Horrela, *covidImpact*>50 handiagoko sintomak soilik hartuko dira kontuan.

Gero, antzekoa egin dut *Increment* klasean, baina kasu honetan beharrezkoa da pazientearen urteak eta *afection*-a zenbatekoa diren jakitea. Horrela, pazientearen urteeraren arabera *Increment* klaseko metodoak pazientearen inkrementua kalkulatu ahalko du beharrezko if-ekin. Honela, *Covid19Pacient* klaseko *calcCovid19Impact* metodoa inpaktua kalkulatzeari bakarrik arduratuko da.

Azkenik calcCovid19Impact metodoak klase berriei deitzeko, hauen instantzia bat sortu dut metodoan bertan.

3. Zein aldaketa egin beharko litzateke 45 eta 65-en arteko pazientei ere %30 inkrementua aplikatzeko?

Increment klaseko getImpact_yearsIncrement metodoan beste if bat jarri beharko da non baldintza berria gertatzen bada increment berria kalkulatu duen. Gure kasuan:

```
if(pacientYears>45 && pacientYears<65) {increment=afection*0.3;}
```

4. Zein aldaketa egin beharko litzateke, 10 baino gazteagoko pazientei beti 0 inpaktua itzultzeko?

Covid19Pacient klaseko calcCovid19Impact metodoaren hasieran 10 urte baino gutxiagoko pertsona den konprobaketa egin dezakegu eta horrela bada 0 itzuli.

Liskov printzipioa (LSK)

Galderak:

1. Programa batean, Covid19Pacient bat sortu sintoma desberdin batzuekin, eta jarraian bere metodoak exekutatu.

```
public class _1_Pacient_LSK_Proba {
    public static void main(String[] args) {
        Covid19Pacient pazientea= new Covid19Pacient("Julen_HiruSintoma", 21);

        int cardiImpact = 2; int cardiIndex=1; int cardioAffecteddays=3;
        CardioVascularSymptom cardiovascular= new
        CardioVascularSymptom("Cardioname", cardiImpact,cardiIndex);

        int neurolImpact = 3; int neurolIndex=2;
        NeuroMuscularSymptom neuromuscular= new
        NeuroMuscularSymptom("Neuroname",neurolImpact, neurolIndex);

        int respiImpact = 1; int respiIndex=1;
        RespiratorySymptom respiratory = new
        RespiratorySymptom("respirname", respiImpact,respiIndex);

        int incurImpact = 3; int incurIndex=3;
        IncurableSymptom incurable= new
        IncurableSymptom("incurname",incurImpact, incurIndex);
```

```

    pazientea.addSymptom(cardiovascular,70);
    pazientea.addSymptom(neuromuscular,80 );
    pazientea.addSymptom(respiratory,60 );
    pazientea.addSymptom(incurable,90 );

    pazientea.showSymptoms();
    pazientea.cure();
  }

```

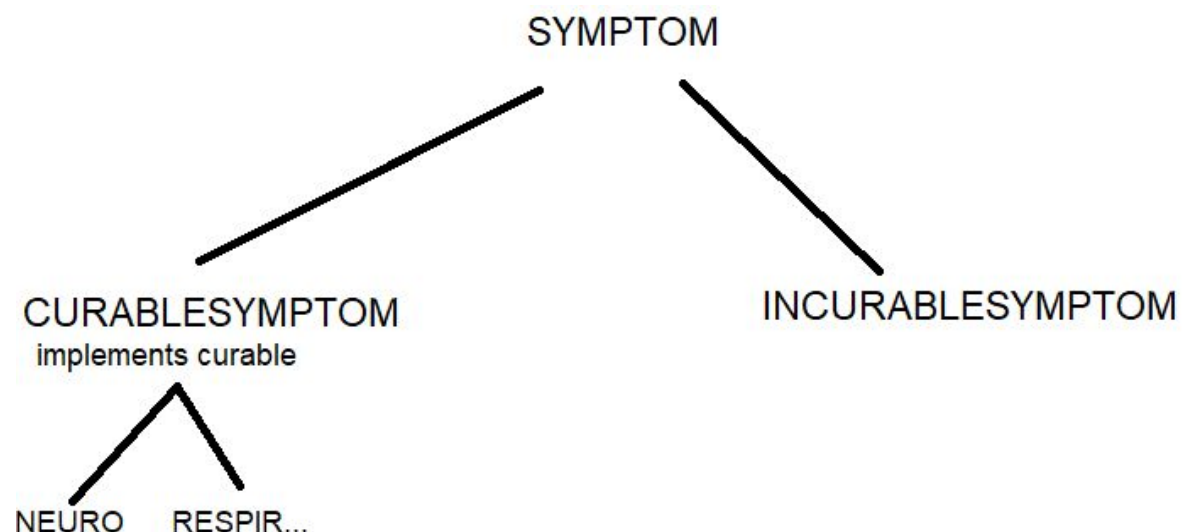
2. *Project klaseak OCP printzipioa betetzen du?. Justifikatu erantzuna.*

OCP printzipioa betetzen du, hierarkikoki ezarria daudelako sintomak eta kode berria hierarkian kode berria idaztearen bera izango litzatekeelako.

3. *Project klaseak LSK (Liskov) printzipioa betetzen du?. Justifikatu erantzuna.*

Ez, Hierarkiaren goikoa symptom klasea da eta honek zuzenean 2 metodo erabiltzen ditu. Hauek zabaldu eta interfazeen erabilera izan ordez.

4. *ErrefaktORIZATU aplikazioa OCP edo LSK betetzen ez badu.*



Lehenik honako eskematxoa egin dut nondik norakoa planteatzearen. Hau bistan izanda, argi geratzen zait Curablesympton eta incurablesympton klaseek metodo konpartitua izango dutela (Symptom klaseko show()). Hori dela eta, CurableSymptom klaseek bakarrik “zaindu” (cure) ahalko du eta hau implementatu beharko du honen umeek zaindu ahal izateko. IncurableSymptom klasearen kasuan, ez zaigu planteatzen hauetako sintomaren adibiderik. Horregatik ez dut umerik sortu.

Azkenik Covid19Pacient klasean cure() metodoa aldatu beharko genuke, zaindu edo cure CurableSymptom klasearen instantziek bakarrik egingo dutelako.

Dependentzi inbertsioaren printzipioa (DIP)

Galderak:

1. DIP printzipioa betetzen du. Justifikatu erantzuna. Hau da, posible da SRP ariketan sortu diren afekzio eta inkrementu portaera desberdinak aplikatu, Covid19Pacient klasea aldatu gabe?

Ez du DIP printzipioa betetzen. Covid19Pacient klaseak CalcAfection eta CalcIncrement klaseak atzitu ditzazke. Hori dela eta, bertan abstrakzio maila bat sortu beharko litzateke. Horrela, klase berriak sortzean interfaze berriak hedatuz. Gainera kodea gehitu beharko litzateke, hashmap-a, eraikitzaile falta...

2. ErrefaktORIZATU kodea betetzen ez badu.

AfectionABSTRACTION eta IncrementABSTRACTION klase abstraktuak sortu ditut. Hori dela eta, Covid19Pacient-eko calcCovid19Impact metodoak input moduan klase abstraktu hauek izan beharko ditu (exekutatzen ari den umea zein den jakin gabe). Orain, gure CalcAfection eta CalcIncrement klaseek klase abstraktu berriak hedatuko dituzte eta hauen antzera klase berriak sortu ahalko ditugu dagokien abstrakzioen hedapenarekin.

3. Programa nagusi bat adibide batekin sortu.

```
package DIP;
public class HIRUsintomako_pazientea {

    public static void main(String[] args) {
        Covid19Pacient hiru_sintomako_pazientea= new
Covid19Pacient("Julen_HiruSintoma", 21);

        int cardiImpact = 50; int cardiIndex=1; int cardioAffecteddays=3;
        CardioVascularSymptom cardiovascular= new
CardioVascularSymptom(cardiImpact,cardiIndex,cardioAffecteddays);

        int neuroImpact = 30; int neuroIndex=2; int neuroAffecteddays=3;
        NeuroMuscularSymptom neuromuscular= new
NeuroMuscularSymptom(neuroImpact, neuroIndex,neuroAffecteddays);

        int respirImpact = 70; int respirIndex=3; int respirAffecteddays=3;
```

```

        RespiratorySymptom respiratory = new
RespiratorySymptom(respirImpact,respirIndex,respirAffecteddays);

        hiru_sintomako_pazientea.addSymptom(cardiovascular,90);
        hiru_sintomako_pazientea.addSymptom(neuromuscular,90 );
        hiru_sintomako_pazientea.addSymptom(respiratory,90 );

        AfecctionABSTRACTION afec= new CalcAfecction();
        IncrementABSTRACTION inc= new CalcIncrement();

        double
covid_impact=hiru_sintomako_pazientea.calcCovid19Impact(afec, inc);
        System.out.println("The covid impact of
"+hiru_sintomako_pazientea.getName() +" is: "+covid_impact);
    }}

```

Interfaze Bananduaren Printzipioa (ISP).

1. Zer informazio behar du CalcIncrement klaseak egin behar duen funtzionalitatea betetzeko, eta zer informazio jasotzen du? ISP printzipioa bortxatzen dutela uste duzu?.

Printzipio honetan “bezeroak” erabiliko dituen printzipioak bakarrik izan behar ditu erabilgarri eta gure kasuan pazientearen klasea pasatzen badiogu CalcIncrement klaseari metodo gehiago izango ditugu erabilgarri. CalcIncrement klaseari Paziente bat pasatzen dugun bakoitzean ondo funtzionatuko du programak baina yearable interfaze bat egin beharko genuke getyears metodoa bakarrik atzitu ahal izateko.

2. CalcIncrement klasea errefaktORIZATU, Pacient parametroa aldatuz interface batengatik. Covid19Pacient klasea ere aldatu.

Pacient klaseari yearable interfazea inplementatzeko esan diot. Horrela, pazietea pasatzean parametro gisa yearable dela ikusiko du eta ondo funtzionatuko du. Baina beste metodoei atzipena izan gabe.

3. Pertsona klase berri bat emanda, zer aldaketa egin beharko zenuke Pertsona klasean, calculateIncrementPerYear metodoari deitzeko Pertsona objektu batekin (eta indize afekzio) bere inkrementua kalkulatzeke. public class Pertsona { String name, adina; } Programa bat sortu, CalcIncrement klaseko calculateIncrement metodoari deitzen diona Pertsona objektu batekin, eta 40ko afecction indizearekin.

Hasteko suposatu dut String adina ordeztu dela eta aldatu dut. Ondoren, getter eta setterak sortu ditut eta getAdina() metodoaren izena getYears-engatik aldatu dut. Kasu honetan yearable izan beharko du klaseak CalcIncrement klaseko metodoak metodo bera .getYears() erabiltzeko. Azkenik, MAIN_getIncrement klasean pertsonen eta inkrementatzeko klasearen instantziak egin eta soluzioa idatzi dut.