16-4-2018

Denda bat kudeatzeko aplikazioa

Programazioa

3. BERTSIOA

Fitxategiak erabilita

Oihane Axpe Telleriarte
DAM 1-eko ikaslea

IES Uni Eibar-Ermua
Eibar, Gipuzkoa

AURKIBIDEA

IRUDIEN TAULA

TAULAK

1. SARRERA

Sortutako aplikazioa denda bat kudeatzeko sortu da.

Proiektu hau, Netbeans 8.2-rekin sortuta dago, java lengoaian idatzita.

Hau, bigarren bertsioa izango da eta dokumentu honek aldaketak jasan dezake edozein puntutan, proiektua aurrera doan heinean.

Dokumentu honek, honako estruktura hau dauka: Lehenengo, proiektuaren diagrama klasea nolakoa izango den azaltzen da. Ondoren, proiektuaren egitura nolakoa izango den azalduko da, hau da, klaseak, atributuak, herentzia, metodoak eta metodo estatikoak, getter eta setter-ak, array-ak, fitxategiak eta erroreen kontrola nola kudeatuko den. Gainera, kontsolaren nondik norakoak ere azalduko dira.

2. PROIEKTUA

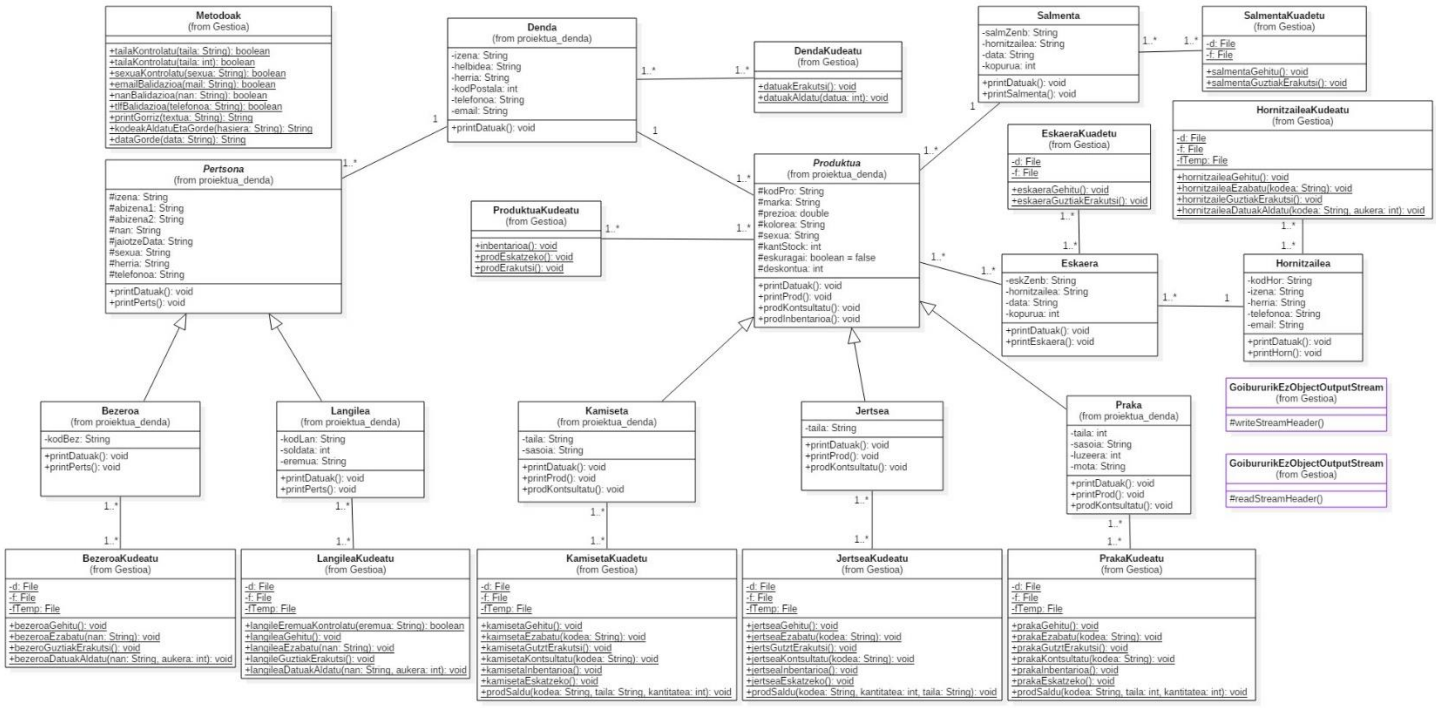
Aplikazio honek, denda bat gestionatzeko balio du. Dendan lan egingo duten langileak gestionatzeko aukera emango du. Bezeroak ere kudeatzeko aukera egongo da, eta salduko diren produktuak kudeatzeko ere erabiliko da, baita hornitzaileei produktuak eskatzeko ere.

2.1. Diagrama

Proiektua kodifikatzen hasi aurretik, diagrama bat egin da, edukiko dituen klaseak, herentzia eta klaseen arteko erlazioak kontuan hartuta.

Ondorengo diagraman, proiektuak izango dituen klase guztiak ikusi daitezke.

Klase bakoitzak, atributuak eta metodoak dituzte, ondorengo irudian ikusten den bezala.



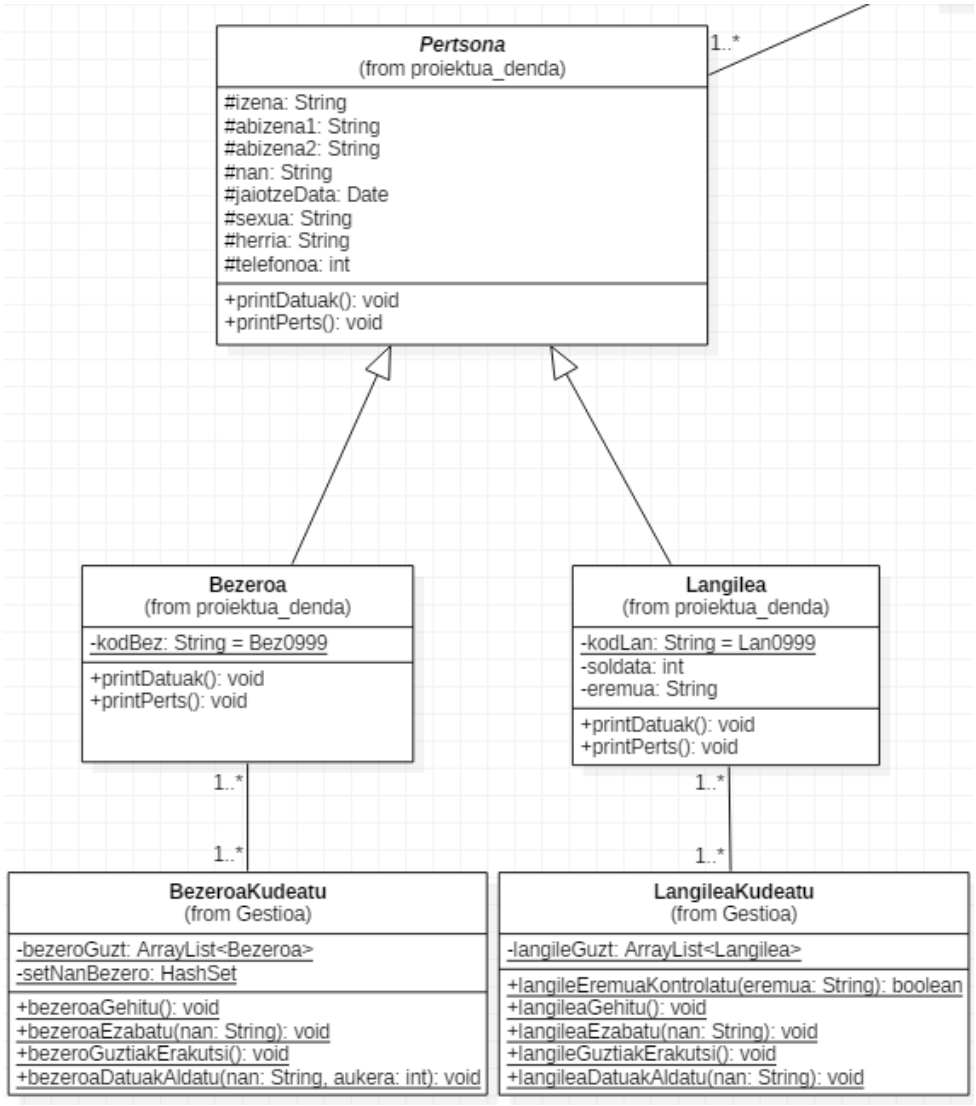
1. Irudia: Proiektuaren diagrama

Horrez gain, getter eta setter-ak ere izango dituzte. Hauek, metodo publikoak dira eta atributuetara heltzeko balio dute. Aurrerago (Getter & Setter puntuan), adibideekin batera azalduko dira gehiago.

Diagraman ikusten den bezala, proiektuak bi atal nagusi izango ditu. Alde batetik, bezero eta langileen kudeaketa eta bestetik, produktuen kudeaketa.

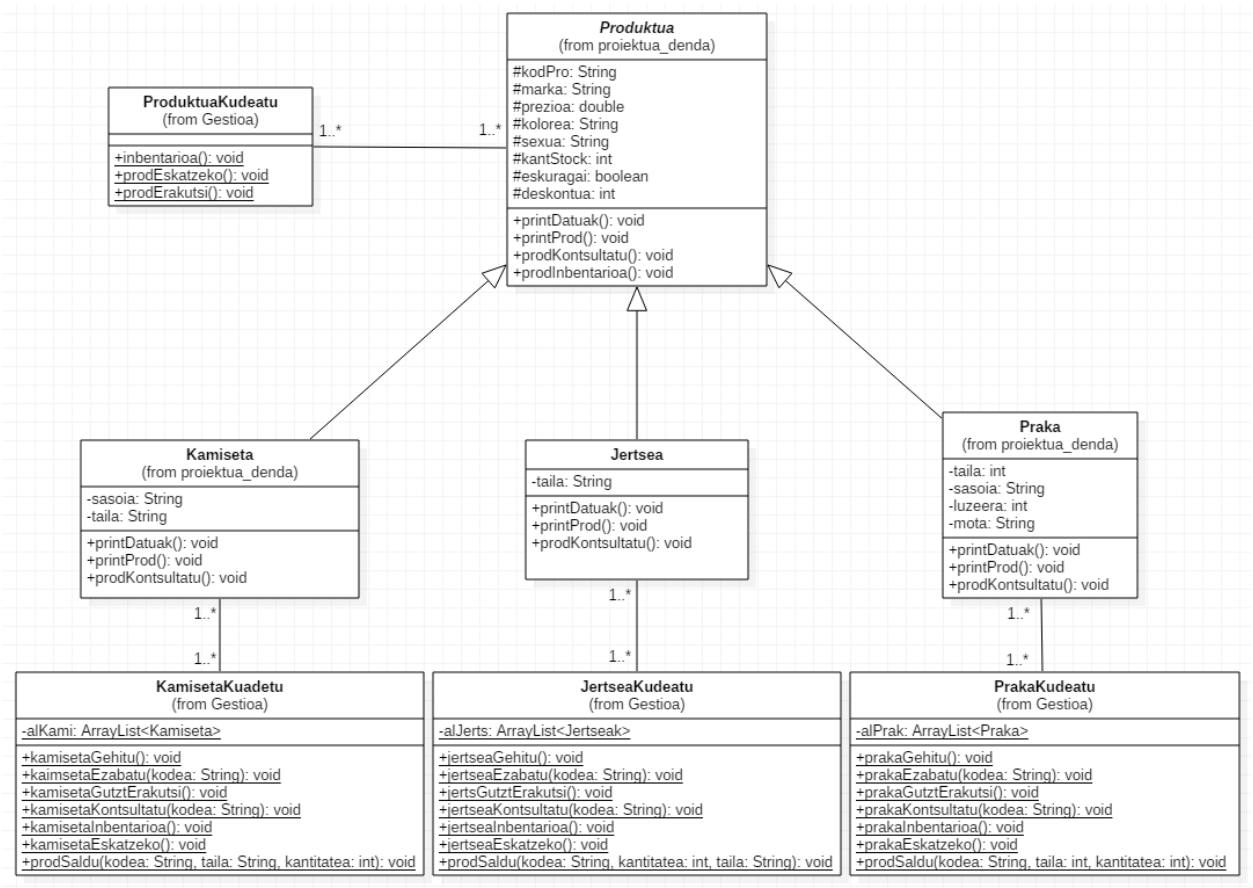
Proiektu honetan, bi herentzia izango ditugu. Bat pertsona eta bezero-langileen artean eta bestea, produktuena.

1. Pertsona klaseak, bezeroa eta langilea azpi klaseak ditu.



2. Irudia: Herentzia - Pertsona

2. Produktuak klaseak, kamiseta, jertseak eta prakak azpi klaseak ditu.



3. Irudia: Herentzia - Produktuak

Klaseetako atributu eta metodoak, publikoak, pribatuak edo babestuak izan daitezke, eta lortu nahi dugunaren araberakoak izango dira.

Ondorengo taulan ikusi daitezke modifikatzaileen desberdintasunak.

+	**public**	Edozein klaseetatik ikusi daiteke.
-	**private**	Klasetik kanpora ezin da ikusi.
#	**protected**	Klasetik bertatik eta azpi klaseetatik bakarrik ikusi daiteke.

1. Taula: Modifikatzaileen desberdintasunak

Klaseen arteko erlazioa adierazteko:

 Herentzia. Super klase eta azpi klaseen arteko erlazioa.

 Klaseen arteko erlazioa adierazten du.

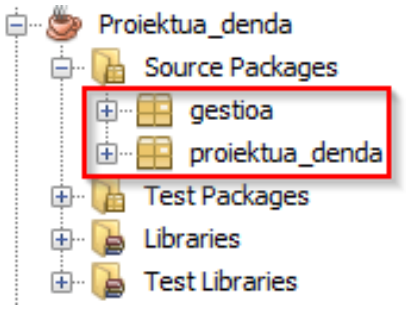
3. PROIEKTUAREN EGITURA

Aurreko puntuan ikusitako klase diagrama kontuan hartuta, klaseak sortu dira. Atal honetan, klaseak, atributuak, metodoak… zeintzuk diren eta nola dauden sortuta azaltzen da.

3.1. Klaseak

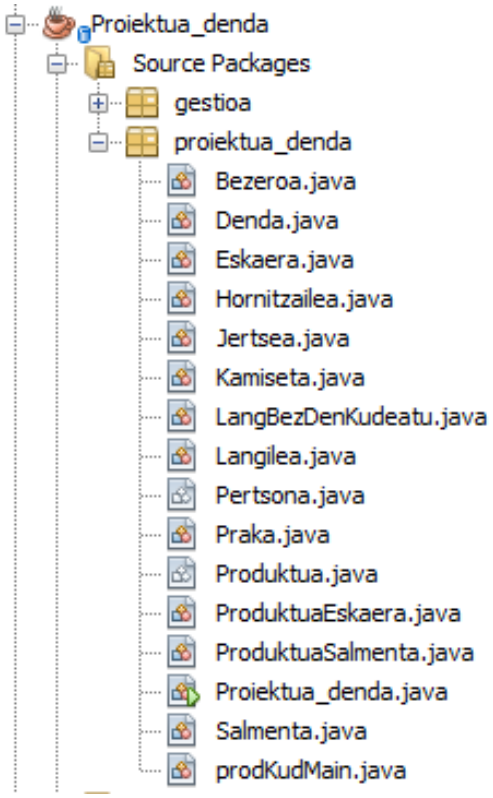
Honako hauek dira proiektu honetan sortutako klaseak.

Bi pakete sortu dira, ***gestioa*** eta ***proiektua_denda***, ordena mantentzeko.



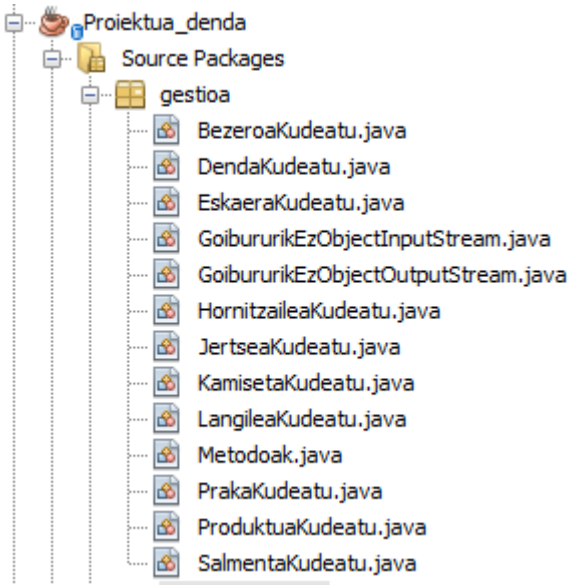
4. Irudia: Paketeak

"proiektua_denda" paketen, proiektuko klaseak daude.



5. Irudia: Klaseak

"Gestioa" paketean berriz, beste paketeko klaseak kudeatzeko sortu diren klaseak daude, non objektu bakoitza kudeatzeko metodo estatikoak dauden.



6. Irudia: Kudeatzeko klaseak

Kolore desberdinez agertzen direnak (Produktuak.java adibidez), **klase abstraktuak** dira. Klase hauen instantzia berririk ezingo da sortu. Hau lortzeko, kodean "*abstract*" hitza erabili behar da. Hona hemen adibide bat.

```
public abstract class Produktuak {
```

7. Irudia: Klase abstraktuak

Proiektua kudeatzeko klaseak ("proiektua_denda" paketean daude):

- **Proiektua_denda.java** → main. Proiektua_denda.java
- **LangBezDenKudeatu.java** → *langBezDenKudeatu()* funtzioa. Mainetik deitu. Honek langileak, bezeroak eta denda kudeatzeko aukera ematen du.
- **prodKudMain.java** → bi funtzio nagusi ditu definituta, *prodKudMain()* eta *horniKudMain().*
    - *prodKudMain()* funtzioa. Mainetik deitu. Honek, produktu desberdinak kudeatzeko aukera ematen du.
    - *horniKudMain()* funtzioa. Mainetik deitu. Honek, hornitzaileak kudeatzeko aukera ematen du.

Hauetan, funtzio desberdinak egongo dira (menuak adibidez).

3.2. Herentzia

Herentzia, super klase bat sortzea da, honen azpi klaseak, bere atributu bai metodoak heredatzeko asmoz. Horrela, azpi klaseetan, ez lirateke egongo atributu eta metodoak errepikatuta.

Esan bezala, klase bat heredatzerakoan, klase horrek duen atributu bai metodoak heredatzen dira. Hau da, superklaseak dituen atributu eta metodoak, azpi klaseek ere izango dute.

Hau lortzeko, kodean "*extends*" hitza erabili behar da. Hona hemen adibide bat.

```
public class Kamiseta extends Produktuak {
```

8. Irudia: Herentzia

Adibide horretan, Kamiseta, Produktuak klasearen azpi klasea da. Produktuak dituen atributu eta metodoak heredatuko ditu.

Honekin, kodea klaseetan behin eta berriz errepikatzea saihestuko dugu, amankomunean daukaten atributu eta metodoak aita izango den klasean jarriz.

Proiektu honetan, herentzia behin baino gehiagotan aplikatzen da. Honen adibide bat, pertsona, langile eta bezeroen artekoa da.

Langile eta bezero guztiek, izena, abizenak, NAN-a, jaiotze data… izango dute. Atributu guzti horiek behin eta berriz errepikatu ordez, **Pertsona** izeneko klase bat sortu eta atributu honek bertan jarri daitezke, ondorengo argazkian ikusten den bezala.

```
public abstract class Pertsona implements Serializable {
    /* ATRIBUTOAK */
    protected String izena;
    protected String abizenal;
    protected String abizena2;
    protected String nan;
    protected String jaiotzeData;
    protected String sexua;
    protected String herria;
    protected String telefonoa;
```

9. Irudia: Adibidea - Herentzia Pertsona

Pertsona klaseak atributu guzti horiek izanik, eta Langilea eta Bezeroa klaseetan herentzia aplikatuz, hauetan desberdinak direnak bakarrik jarri beharko lirateke. Hona hemen bi adibideak.

1. Langilea klasean, kodLan (langilearen kodea), soldata eta eremua daude.

```
public class Langilea extends Pertsona implements Serializable {
    /* ATRIBUTOAK */
    private String kodLan;
    private double soldata;
    private String eremua; // saltzailea edo garbitzailea den gorde
```

10. Irudia: Adibidea - Herentzia Langilea

2. Bezeroa klasean, berriz, kodBez (bezeroaren kodea).

```
public class Bezeroa extends Pertsona implements Serializable {
    /* ATRIBUTOAK */
    private String kodBez;
```

11. Irudia: Adibidea - Herentzia Bezeroa

3.3. Atributuak

Klase bakoitzak, beharrezkoak dituen atributuak ditu. Atributuak definitzerakoan, zein modifikatzaile izango duen (public, private, protected edo default/package), zein datu mota izango den (int, String, double…) eta edukiko duen izena jarri behar dira.

```
/* ATRIBUTOAK */
private String kodHor;
private String izena;
private String herria;
private String telefonoa;
private String email;
```

12. Irudia: Atributuak

Atributu guztiak, pribatuak (private) izango dira, salbuespen batekin. Herentzia aplikatuta dauden azpiklaseetan, babestuak (protected) izango dira.

```
/* ATRIBUTOAK */
protected String kodPro;
protected String marka;
protected double prezioa;
protected String kolorea;
protected String sexua;
protected int kantStock;
protected boolean eskuragai=false;
protected int deskontua=0;
```

13. Irudia: Atributuak (herentziarekin)

3.4. Eraikitzaileak

Eraikitzaileak, objektu berri bat sortzeko balio dute. Hauen izena, klasearen berdina izan behar da eta ez dute ezer bueltatzen.

Objektu baten instantzia berri bat egiten denean, eraikitzaileari deitzen zaio. Instantzia berri bat egiteko, *"new"* hitza erabiltzen da.

```
Denda dendal = new Denda();
```

14. Irudia: Adibidea – Instantzia berria sortu

Klase berdin bakoitzean, eraikitzaileen gainkarga egon daiteke. Guztiek izen berdina izan behar dute, baina, parametro desberdinak. Hau da, parametro kopurua eta ordena desberdina izan behar da.

```
/* ERAIKITZAILEAK */
public Kamiseta () {...5 lines }

public Kamiseta (String kodea, String marka, double prezioa, String kolorea, String sexua, int kantStock, String taila, String sasoia) {...5 lines }

public Kamiseta (String kodea, String sexua, double prezioa, String sasoia) {...4 lines }
```

15. Irudia: Eraikitzaileetan gainkarga

Eraikitzaileen adibide bat, Kamiseta klaseko hau izan daiteke. Hemen, aurretik esan bezala, gainkarga daukagu.

```
/* ERAIKITZAILEAK */
public Kamiseta () {
    super();
    setTaila();
    setSasoia();
}

public Kamiseta (String kodea, String sexua, double prezioa, String sasoia) {
    super(kodea, sexua, prezioa);
    this.sasoia=sasoia;
}
```

16. Irudia: Adibidea - Eraikitzaileak

OHARRA: super(); edo ***super(arg1, arg2…);*** erabiltzen dira, herentzia dutenen kasuan, super klaseko eraikitzaileari deitzeko.

Objektu berri bat sortzeko, instantzia berri bat egiten da, eraikitzaile bati deituz. Parametro kopuru eta ordenaren arabera, eraikitzaile bati edo beste bati deitzen zaio.

```
/* PRODUKTUAK SORTU */
Jertseak jertsGizl = new Jertseak("1206596-8223", "Ternua", 44.99, "Beltza", "Gizona", 15, "M");
Kamiseta kamiGizl = new Kamiseta("CE5205", "Adidas", 24.99, "Urdina", "Gizona", 15, "L", "Uda");
Prakak prakGizl = new Prakak("1273283-9937", "Ternua", 99.99, "Beltza", "Gizona", 15, 38);
```

17. Irudia: Adibidea - Objektu berria

3.5. Metodoak

Metodoek, definituta daukaten zeregin jakin bat egiten dute eta izenaren bidez deitzen dira.

Herentziadun klaseetan, azpi klaseetan, ***super.<MetodoIzena>()*** erabili beharko da superrari deitu eta datu guztiak hartzeko, bai azpi klaseak duena eta baita superklaseak duenak ere.

Ondorengo irudian ikusi daiteke adibide argi bat. Lehenengo irudiko metodoa, **Kamiseta** klasean definituta dago. Honek, **Produktua** klaseko atributuak hartzen dituenez, *super.printDatuak()* jarri behar da. *super.printDatuak()*, superklasearen *printDatuak()* metodoari deituko dio (ikusi gezi gorria) eta bertakoa inprimatu, bestearekin batera.

```
/* METODOAK */
@Override
public void printDatuak() {
    super.printDatuak();
    System.out.println("Taila: "+taila);
    System.out.println("Sasoia: "+sasoia);
}
```

18. Irudia: Adibidea - printDatuak() metodoa (Kamiseta)

```
/* METODOAK */
public void printDatuak() {
    System.out.println("\nKodea: "+kodPro);
    System.out.println("Marka: "+marka);
    System.out.println("Prezioa : "+prezioa+"€");
    System.out.println("Kolorea: "+kolorea);
    System.out.println("Sexua: "+sexua);
    System.out.println("Stock-ean: "+kantStock);
}
```

19. Irudia: Adibidea - printDatuak() metodoa (Produktua)

Metodo horrek, honako hau bueltatuko luke:

```
Kodea: CE5205
Marka: Adidas
Prezioa : 24.99€
Kolorea: Urdina
Sexua: Gizona
Stock-ean: 15
Taila: S
Sasoia: Uda
```

20. Irudia: Adibidea - printDatuak()

printDatuak() metodoa klase guztietan sortu da. Honek, klaseko atributu guztien datuak erakusten ditu.

printDatuak() metodoaz gain, beste hainbat sortu dira. Ondorengo irudian ikusi daitezke adibide batzuk. Metodo hauek, Kamiseta klasean definituta daudenak dira. Hauek, ondoren, Kamiseta kudeatzeko sortuta dauden beste klasean erabiliko dira, datu zehatz batzuk bistaratzeko.

```
/* METODOAK */
@Override
public void printDatuak() {
    super.printDatuak();
    System.out.println("Taila: "+this.taila);
    System.out.println("Sasoia: "+this.sasoia);
}

@Override
public void printProd() {
    super.printProd();
    System.out.printf("    %1$-10s    %2$-10s\n", this.taila, this.sasoia);
}

@Override
public void prodKontsultatu() {
    super.prodKontsultatu();
    System.out.printf("    %1$-10s    %2$-10s\n", this.taila, getKantStock());
}
```

21. Irudia: Adibidea - Metodoak

Datuak pantailan inprimitzerakoan, zutabetan edo errenkadetan ikusi nahi denaren arabera, informazioa nahasi zamar agertu daiteke.

Datuak errenkadetan bistaratzeko, **System.out.printf()** metodoa erabili da, datuen arteko espazioa kontrolatzeko.

Ondoko irudian, horren adibide bat.

```
@Override
public void prodKontsultatu() {
    super.prodKontsultatu();
    System.out.printf("    %1$-10s    %2$-10s\n", this.taila, getKantStock());
}
```

22. Irudia: Adibidea - prodKontsultatu() metodoa (Kamiseta)

```
public void prodKontsultatu() {
    System.out.printf("%1$-15s    %2$-10s", this.kodPro, this.kolorea);
}
```

23. Irudia: Adibidea - prodKontsultatu() metodoa (Produktua)

%1$-15s bezalako parametroak erabiltzen dira. “%1$” zatiak, zenbatgarren aldagaiari egiten dion erreferentzia markatzen du. “15s” zatiak berriz, datuen arteko hutsune kopurua markatzen du.

```
System.out.printf("%1$-15s     %2$-10s", this.kodPro, this.kolorea);
```

Espazioak kontrolatuta, ondorengo irudian ikusten den emaitza lortuko da.

Kodea	Kolorea	Taila	Kantitatea
212103-524	Berdea	XS	10
212103-524	Berdea	M	10
212103-524	Berdea	L	10

24. Irudia: Adibidea- prodKontsultatu()

3.5.1. Metodo estatikoak

Kamisetak, prakak, jertseak, bezeroak… hauetako klase kudeatzeko, beste klase bat sortu da, **KamisetaKudeatu.java** bezalakoak. Kudeatzeko klase hauetako bakoitzean, fitxategi bat sortu da objektuak bertan gordetzeko eta metodo estatikoak, objektuak kudeatzeko. Kudeaketa hori, fitxategien gainean egiten da, aurrerago ikusiko den bezala.

Metodo estatikok definitzeko, *“static”* hitza erabiltzen da.

```
public static void kamisetaGehitu() {
```

25. Irudia: Adibidea - Metodo estatiko bat definitu

Metodo hauek, instantzia berririk egin gabe erabili daitezke.

```
KamisetaKudeatu.kamisetaGehitu();
```

26. Irudia: Metodo estatikoak erabiltzen

Irudian ikusten den bezala, kamisetaGehitu() metodo estatikoa da, eta metodo hori erabiltzeko, ez dago instantziatu beharrik. Nahikoa da metodoa definituta dagoen klasearen izena aurretik jartzea.

<Klase_izena>.<metodo_estatiko_izena>;

Metodo estatikoen adibide batzuk, hondoko irudian ikusten da. Aurretik komentatu bezala, objektuak gordetzeko fitxategi bat sortu da eta objektu horiek kudeatzeko (gehitu, ezabatu, erakutsi…) metodo estatikoak.

Fitxategiak, Objektuak karpeta barruan sortzen dira eta **kamiseta.obj** bezalakoak izango dira.

```
public class KamisetaKudeatu {
    private static File d = new File("Objektuak");
    private static File f = new File(d+"\\kamiseta.obj");
    private static File fTemp = new File(d+"\\kamiTemp.obj");

    /* Kamiseta berri bat gehitu */
    public static void kamisetaGehitu() {...21 lines }

    /* Kamiseta zehatz baten datu guztiak ezabatu */
    public static void kamisetaEzabatu(String kodea) throws IOException {...32 lines }

    /* ArrayList-eko Kamiseta guztien datuak erakusteko metodoa */
    public static void kamisetaGutztErakutsi() throws IOException {...24 lines }

    /* Kamiseta baten kodea, ArrayList-ean dagoen kontsultatu, dendan dagoen jakiteko. */
    public static void kamisetaKontsultatu(String kodea) {...25 lines }

    /* Dauden kamiseta guztiak erakusteko metodoa */
    public static void kamisetaInbentarioa() {...22 lines }

    /* kantitatea 5 baino gutxiago duten kamisetak erakusten ditu */
    public static void kamisetaEskatzeko() {...24 lines }

    /* KAMISETAK saltzeko metodoa. Erabiltzaileak kodea, taila eta kantitatea sartuko ditu. */
    public static void prodSaldu(String kodea, String taila, int kantitatea) {...57 lines }

}
```

27. Irudia: Metodo estatikoak

Metodo estatiko baten adibidea ondoko irudian ikusi daiteke. Metodo hau, objektu bat (kamiseta kasu honetan) fitxategira gehitzeko sortu da.

1. Objektuaren instantzia berria sortzen da.
2. Sortutako instantzia berri hori fitxategian idatzi. Honetarako, ***geoos.write(kami1);*** erabiltzen da. Aurrerago azalduko da fitxategien funtzionamendua.

```
/* Kamiseta berri bat gehitu */
public static void kamisetaGehitu() {
    if (!d.exists()) {
        d.mkdir();
    }
    try {
        GoibururikEzObjectOutputStream geoos = new GoibururikEzObjectOutputStream(new FileOutputStream(f, true));
        System.out.println("Kamiseta berriaren datuak sartu behar dituzu.\n");
        Kamiseta kamil = new Kamiseta();
        geoos.writeObject(kamil); // objektua fitxategian idatzi
        geoos.flush();
        geoos.close();
        System.out.println();
        System.out.println("Datu hauek dituen produktua gorde da."
                + "\nProduktua: KAMISETA");
        kamil.printDatuak();
    } catch (FileNotFoundException ex) {
        System.out.println(Metodoak.printGorriz("Fitxategia ez du aurkitzen!"));
    } catch (IOException ex) {
        System.out.println(Metodoak.printGorriz("Arazoak daude datuak jasotzerakoan"));
    }
}
```

28. Irudia: Adibidea - Metodo estatikoa

Bezero, langile, hornitzaile, eskaera eta salmenta kodeak erregistratzeko orduan, automatikoki erregistratzen dira. Horretarako, **kodeak.txt** fitxategi bat sortu da, non kode hauek hasieratu diren.

Ondorengo irudian ikusi daiteke hauen kontrola eramateko adibidea. Kontrol hori eramateko, metodo estatiko bat sortu da.

```
public static String kodeakAldatuEtaGorde(String hasiera) {
    FileReader fr = null;
    BufferedReader br = null;
    FileWriter fw = null;
    PrintWriter pw = null;
    String kodea = null;
    File f = new File("kodeak.txt"); // kodeak gordeta dauden fitxategia
    hasiera = hasiera.toLowerCase(); // minuskulaz
    try {
        ArrayList<String> kodGuztiak = new ArrayList<String>(); // fitxategiko kode guztiak arraylistean gordetzeko
        fr = new FileReader(f);
        br = new BufferedReader(fr);
        String lerroa;

        while((lerroa=br.readLine()) != null){
            if (lerroa.contains(hasiera)) { //kodearen hasiera konprobatu
                kodea = String.valueOf(lerroa.substring(0, hasiera.length()+1) + (Integer.parseInt(lerroa.substring(hasiera.length()+1)) +1));
                lerroa = lerroa.replace(lerroa, kodea);
            }
            kodGuztiak.add(lerroa); // lerroa arraylistean gorde
        }

        fw = new FileWriter(f);
        pw = new PrintWriter(fw);

        for (int i = 0; i<kodGuztiak.size(); i++) {
            pw.println(kodGuztiak.get(i)); // fitxategian kode guztiak berriz idatzi (aldaketa eginda)
            pw.flush();
        }
    } catch (FileNotFoundException ex) {
        System.out.println("Fitxategia ez da existitzen.");
    } catch (IOException ex) {
        Logger.getLogger(Metodoak.class.getName()).log(Level.SEVERE, null, ex);
    }
    finally {
        try {
            fr.close();
            br.close();
            pw.flush();
            fw.close();
            pw.close();
        } catch (IOException ex) {
            Logger.getLogger(Metodoak.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    return kodea; // kode zenbakia bueltatu, objektuan gordetzeko
}
```

29. Irudia: kodeakAldatuEtagorde(String hasiera): String metodoa

Fitxategian dagoena irakurtzen du eta bere lerroa bilatzen duenean, kodearen azkenengo lau karaktereak hartzen ditu, hauek zenbaki bihurtu eta +1 egin. Ondoren, berriz eraikitzen da String-a eta fitxategian gorde.

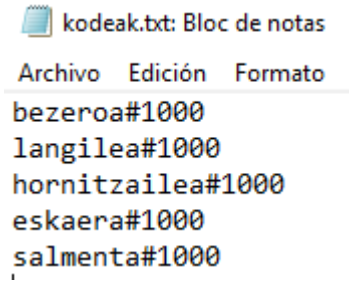
Ukitzen ez diren kodeak ez galtzeko, kodeak aldatzen hasi aurretik, arraylist batean gordetzen dira.

Metodo estatiko hau, bezero, langile, eskaera, hornitzaile eta salmenta klaseetan erabiliko da, klase bakoitzeko kodearen set metodoan hain zuzen ere.

```
public void setKodBez() {
    this.kodBez = Metodoak.kodeakAldatuEtaGorde("Bezeroa");
}
```

30. Irudia: Adibidea - Getter & Setter (kodea automatikoki gehitzen)

Hauen defektuzko balioak honako hauek dira:



31. Irudia: kodeak.txt fitxategiko balioak

Sortu diren metodo estatikoak honako hauek dira:

Metodoak klasea (Metodoak.java)

- ✓ **printGorriz(textua: String): String** → Textua gorriz bueltatzen duen metodoa.
- ✓ **tailaKontrolatu(taila: String): boolean** eta **tailaKontrolatu(taila: int): boolean** → produktuen taila kontrolatzeko metodoa. Taila, array-ean dagoen konprobatzen du.
- ✓ **sexuaKontrolatu(sexua: String): boolean** → produktuen edo pertsonan sexua (emakumea edo gizona) kontrolatzeko metodoa. Sexua, array-ean dagoen konprobatzen du.
- ✓ **emailBalidazioa(mail: String): boolean** → Email-a ondo estrukturatuta dagoen konprobatzen duen metodoa. Expresio erregularrak erabiltzen dira (adibidea@adibidea.com)
- ✓ **nanBalidazioa(nan: String): boolean** → NAN zenbakia egokia den edo ez konprobatzen duen metodoa.
- ✓ **tlfBalidazioa(tlf: String): boolean** → Telefono zenbakia egokia den edo ez konprobatzen duen metodoa.
- ✓ **kodeakAldatuEtaGorde(hasiera: String): String** → kodeak.txt fitxategian gordetako kodeak (bezero, langile…) aldatzen dituen metodoa.
- ✓ **dataGorde(data: String): String** → data uuuu/hh/ee formatura bihurtzen duen metodoa.

DendaKudeatu klasea (DendaKudeatu.java)

- ✓ **datuakErakutsi(): void** → Dendaren datuak erakutsi.
- ✓ **datuakAldatu(datua: int): void** → Dendaren datuak aldatu.

BezeroaKudeatu klasea (BezeroaKudeatu.java)

- ✓ **bezeroaGehitu(): void** → bezero berri bat gehitu.
- ✓ **bezeroaEzabatu(nan: String): void** → bezero bat ezabatu, NAN-aren arabera.
- ✓ **bezeroGuztiakErakutsi(): void** → erregistratutako bezero guztiak erakutsi.
- ✓ **bezeroaDatuakAldatu(nan: String, aukera: int): void** → bezero baten datuak aldatu.

LangileaKudeatu klasea (LangileaKudeatu.java)

- ✓ **langileEremuaKontrolatu(eremua: String)** → langileen lan eremua kontrolatzeko metodoa (saltzailea edo garbitzailea izan daitezke). Erabiltzaileak sartutako eremua array-ean dagoen konprobatzen du eta bueltatzen duena, booleano bat da.
- ✓ **langileaGehitu(): void** → langile berri bat gehitu.
- ✓ **langileaEzabatu(nan: String): void** → langile bat ezabatu, NAN-aren arabera.
- ✓ **langileGuztiakErakutsi(): void** → erregistratutako langile guztiak erakutsi.
- ✓ **langileaDatuakAldatu(nan: String, aukera: int): void** → langile baten datuak aldatu.

JertseaKudeatu klasea (JertseaKudeatu.java)

- ✓ **jertseaGehitu(): void** → jertse berri bat gehitu.
- ✓ **jertseaEzabatu(kodea: String): void** → jertse bat ezabatu, kodearen arabera.
- ✓ **jertsGuztErakutsi(): void** → erregistratutako jertse guztiak erakutsi.
- ✓ **jertseaKontsultatu(kodea: String): void** → jertsea, dendan dagoen kontsultatu, kodearen arabera.
- ✓ **jertseaInbentarioa(): void** → dauden jertse guztiak erakutsi.
- ✓ **jertseaEskatzeko(): void** → 5 baino gutxiago dauden jertseak erakutsi.
- ✓ **prodSaldu(kodea: String, kantitatea: int, taila: String): void** → produktuaren salmenta. Erabiltzaileak, kodea, erosi nahi duen kantitatea eta taila sartu beharko ditu.

KamisetaKudeatu klasea (KamisetaKudeatu.java)

- ✓ **kamisetaGehitu(): void** → kamiseta berri bat gehitu.
- ✓ **kamisetaEzabatu(kodea: String): void** → kamiseta bat ezabatu, kodearen arabera.
- ✓ **kamisetaGuztErakutsi(): void**→ erregistratutako kamiseta guztiak erakutsi.
- ✓ **kamisetaKontsultatu(kodea: String): void** → kamiseta, dendan dagoen kontsultatu, kodearen arabera.
- ✓ **kamisetaInbentarioa(): void** → dauden kamiseta guztiak erakutsi.
- ✓ **kamisetaEskatzeko(): void** → 5 baino gutxiago dauden kamisetak erakutsi.
- ✓ **prodSaldu(kodea: String, taila: String, kantitatea: int): void** → produktuaren salmenta. Erabiltzaileak, kodea, taila eta erosi nahi duen kantitatea sartu beharko ditu.

PrakaKudeatu klasea (PrakaKudeatu.java)

- ✓ **prakaGehitu(): void** → praka berri bat gehitu.
- ✓ **prakaEzabatu(kodea: String): void** → praka bat ezabatu, kodearen arabera.
- ✓ **prakaGuztErakutsi(): void** → erregistratuta dauden praka guztiak erakutsi.
- ✓ **prakaKontsultatu(kodea: String): void** → praka, dendan dagoen kontsultatu, kodearen arabera.
- ✓ **prakaInbentarioa(): void** → dauden praka guztiak erakutsi.
- ✓ **prakaEskatzeko(): void** → 5 baino gutxiago dauden prakak erakutsi.
- ✓ **prodSaldu(kodea: String, taila: int, kantitatea: int): void** → produktuaren salmenta. Erabiltzaileak, kodea, taila eta erosi nahi duen kantitatea sartu beharko ditu.

ProduktuaKudeatu klasea (ProduktuaKudeatu.java)

- ✓ **inbentarioa(): void** → dendako produktu guztien inbentarioa, produktuka sailkatuta (kamisetaInbentarioa, jertseaInbentarioa eta prakaInbentarioa erabiltzen dira).
- ✓ **prodEskatzeko(): void** → Stock-ean dauden produktu kantitatea 5 baino txikiagoa diren produktuak erakutsi. Eskaera egiteko

produktuak (kamisetaEskatzeko, jertseaEskatzeko eta prakaEskatzeko erabiltzen dira).

- ✓ **prodErakutsi(): void** → erregistratuta dauden produktu guztiak erakutsi (kamisetaGuztErakutsi, jertsGuztErakutsi eta prakaGuztErakutsi erabiltzen dira).

EskaeraKudeatu klasea (EskaeraKudeatu.java)

- ✓ **eskaeraGehitu(): void** → eskaera berri bat gehitu.
- ✓ **eskaeraGuztiakErakutsi(): void** → erregistratuta dauden eskaera guztiak erakutsi.

SalmentaKudeatu klasea (SalmentaKudeatu.java)

- ✓ **salmentaGehitu(): void** → salmenta berri bat gehitu.
- ✓ **salmentaGuztiakErakutsi(): void** → erregistratuta dauden salmenta guztiak erakutsi.

HornitzaileaKudeatu klasea (HornitzaileaKudeatu.java)

- ✓ **hornitzaileaGehitu(): void** → hornitzaile berri bat gehitu.
- ✓ **hornitzaileaEzabatu(kodea: String): void** → hornitzaile bat ezabatu, kodearen arabera.
- ✓ **hornitzaileGuztiakErakutsi(): void** → erregistratuta dauden hornitzaile guztiak erakutsi.
- ✓ **hornitzaileaDatuakAldatu(kodea: String, aukera: int): void** → Hornitzailearen datuak aldatu.

3.6. Getter & Setter

Aurretik aipatutako metodoez gain, get eta set metodoak ere sortu dira klase guztietan.

- **Getter**-ak, atributu batek duen balioa lortu eta erabiltzeko balio du.
- **Setter**-ak, atributuei balio bat emateko balio dute. Metodo honek, ez du ezer bueltatzen.

Atributu bakoitzeko, set bat eta get bat sortu dira klase bakoitzean. Get-ek, atributuaren balioa bueltatuko du eta set-ek berriz, erabiltzaileak sartu beharko du atributuan gordeko den balioa.

```
public String getKodPro() {...3 lines }
public void setKodPro() {...9 lines }
public String getMarka() {...3 lines }
public void setMarka() {...9 lines }
public double getPrezioa() {...3 lines }
public void setPrezioa() {...9 lines }
public String getKolorea() {...3 lines }
public void setKolorea() {...9 lines }
public String getSexua() {...3 lines }
public void setSexua() {...9 lines }
public int getKantStock() {...3 lines }
public void setKantStock() {...9 lines }
public boolean isEskuragai() {...5 lines }
public void setEskuragai(boolean eskuragai) {...3 lines }
```

32. Irudia: Getter & Setter

OHARRA: Momentuz, set eta get metodoetan, ez dago gainkargarik (adibide bezala, Kamiseta klasean sortu da). Behar izanez gero, aurrerago sortuko lirateke.

```
public void setTaila(String taila) {
    this.taila = taila;
}

public void setSasoia(String sasoia) {
    this.sasoia = sasoia;
}
```

33. Irudia: Adibidea - Set gainkarga

Metodo guzti hauek, antzerakoak izango dira.

1. Get metodoak:

```
public String getTaila() {
    return taila;
}
```

34. Irudia: Adibidea - Getter

2. Set metodoak:

   Erabiltzaileari datuaren balioa sartzeko eskatzen diote. Sartu behar duten datu motaren arabera, datu hori hartzeko modua aldatuko da.

```
public void setTaila() {
    try {
        System.out.print("Sartu taila (S,M,L,XL,XXL): ");
        this.taila = br.readLine();
    }
    catch (IOException gaizki) {
        System.out.println("Arazoak daude datuak sartzerakoan.");
    }
}
```

35. Irudia: Adibidea - Setter

Datak jasotzerako orduan, honako forma honetan egin da. Metodo bat sortu da, data parseatzeko eta *uuuu/hh/ee* formatuan gorde ahal izateko. Metodo honek, **dataGorde** izena dauka.

Ondorengo adibidean ikusten den bezala, data jasotzerako orduan (**setJaiotzeData** metodoan), **dataGorde** metodoari deitzen dio, honek *uuuu/hh/ee* formatuan parseatu eta gordetzeko.

```
public void setJaiotzeData() {
    try {
        System.out.print("Sartu jaiotze data (uuuu/hh/ee): ");
        this.jaiotzeData = Metodoak.dataGorde(br.readLine()); // sartutako data, uuu/hh/ee
    }
    catch (IOException gaizki) {
        System.out.println(Metodoak.printGorriz("Arazoak daude datuak sartzerakoan."));
    }
}

/* String-a data bezela kontrolatzeko metodoa. Data uuuu/hh/ee formatuan bueltatuko du. */
public static String dataGorde(String data) {
    String dataFormatua = null;
    try {
        DateFormat df = new SimpleDateFormat("yyyy/MM/dd");
        Date fetx = df.parse(data); // data hori Date formatura parseatu

        dataFormatua = new SimpleDateFormat("yyyy/MM/dd").format(fetx.getTime()); // formatu zehatz baten jarri
    }
    catch (ParseException gaizki) {
        System.out.println(Metodoak.printGorriz("Ez da kapaza sartutako datuak parseatzeko."));
    }
    return dataFormatua;
}
```

36. Irudia: Adibidea - Getter & Setter (Date mota)

dataGorde metodoan, data jasotzeko formatua definitzen da lehenengo (urtea/hilabetea/eguna) eta ondoren, erabiltzaileak sartu duen string-a ***df.parse(<data>)*** erabilita, datua parseatuko du.

Erroreak kontrolatuta izateko, try eta catch erabiltzen dira (aurrerago azalduko dira).

Datu batzuk kontrolatzen dira, adibidez, email-a eta NAN zenbakia.

```
public void setNan() {
    try {
        do {
            System.out.print("Sartu NAN zenbakia: ");
            this.nan=br.readLine().toUpperCase();
        } while (!Metodoak.nanBalidazioa(nan));
    }
    catch (IOException gaizki) {
        System.out.println("Arazoak daude datuak sartzerakoan.");
    }
}
```

37. Irudia: Adibidea - Getter & Setter (NAN)

Datuen balidazioak egiterakoan, bukle bat sortzen da. Bukletik irtengo da balidazioa betetzen denean. Bitartean, behin eta berriz eskatuko zaio erabiltzaileari datua sartzeko.

3.7. Array-ak

Array-etan, mota berdineko datuak gordetzen dira eta luzeera aurretik zehazten da.

<arrayMota>[] <arrayIzena> = {<definitutakoDatuak>…}

Hiru metodo sortu dira datu zehatz batzuk kontrolatzeko, taila eta sexua hain zuzen. Metodo estatiko hauetako bakoitzean, Array bat definitzen da, eta bertan, datuak zehaztuta egongo dira.

Adibidez, erabiltzaileak produktu baten taila (String motakoa) erregistratzen duenean, kontrol bat eramateko.

```
public static boolean tailaKontrolatu(String taila) {
    String[] tailaKontrola = {"XS", "S", "M", "L", "XL", "XXL"}; // taila posibleak
    boolean aurkituta = Arrays.asList(tailaKontrola).contains(taila.toUpperCase());
    return aurkituta;
}
```

38. Irudia: Adibidea - Array

Array-ak, honako kasuetan erabili dira. Guztietan, atributu zehatz baten datuk kontrolatzeko.

```
public class Metodoak {
    /* produktuen taila kontrolatzeko metodoa. Taila, array-ean dagoen konprobatzen du.
     * Bueltatzen duena booleano bat da.*/
    public static boolean tailaKontrolatu(String taila) {
        String[] tailaKontrola = {"XS", "S", "M", "L", "XL", "XXL"};
        boolean aurkituta = Arrays.asList(tailaKontrola).contains(taila.toUpperCase());
        return aurkituta;
    }
    public static boolean tailaKontrolatu(int taila) {
        int[] arr = {38, 40, 42, 44, 46};
        for (int elementua : arr) {
            if (elementua == taila) {
                return true;
            }
        }
        return false;
    }

    /* produktuen edo pertsonan sexua (emakumea edo gizona) kontrolatzeko metodoa.
     * sexua, array-ean dagoen konprobatzen du.
     * Bueltatzen duena booleano bat da.*/
    public static boolean sexuaKontrolatu(String sexua) {
        String[] sexuaKontrolatu = { "emakumea", "gizona"};
        boolean aurkituta = Arrays.asList(sexuaKontrolatu).contains(sexua.toLowerCase());
        return aurkituta;
    }
}
```

39. Irudia: Array-ak

3.8. Fitxategiak

Fitxategietan, objektuak gordeko dira, ondoren, objektuak bertatik kudeatzeko asmoz. Objektuak gordeta dituen fitxategiak xxxx.obj extentziodunak izango dira.

```
public class Kamiseta extends Produktua implements Serializable {
```

40. Irudia: Adibidea - Serializable

Herentzia daukaten klaseetan, super klaseari ere “implements Serializable” ere jarri behar zaio.

Proiektuan erabiliko diren fitxategi guztiak, honako hauek dira:

- ✓ **bezeroa.obj**: File
- ✓ **langilea.obj:** File
- ✓ **salmenta.obj**: File
- ✓ **eskaera.obj**: File
- ✓ **hornitzailea.obj**: File
- ✓ **kamiseta.obj**: File
- ✓ **jertsea.obj**: File
- ✓ **praka.obj**: File

Fitxategi guzti hauek, Objektuak karpeta barruan sortzen dira.

Fitxategietan objektuak idatzi eta irakurtzeko, eta baita manipulatzeko ere, beste bi klase sortu dira.

- GoibururikEzObjectInputStream
- GoibururikEzObjectOutputStream

Hauekin, objektuak idazterakoan, fitxategian goibururik ez idaztea saihestuko da.

writeStreamHeader() eta readStreamHeader() metodoak berriz idatzi beharko dira, hauek hutsik utzita.

```
public class GoibururikEzObjectOutputStream extends ObjectOutputStream{

    /* ERAIKITZAILEAK */
    public GoibururikEzObjectOutputStream(OutputStream out) throws IOException {
        super(out);
    }

    protected GoibururikEzObjectOutputStream() throws IOException, SecurityException {
        super();
    }

    /* METODOAK */
    /* fitxategiko kabezera ez idazteko metodoa berridatzi, ezer ez egiteko */
    @Override
    protected void writeStreamHeader() throws IOException {
        // metodo honek ez du ezer egiten
    }
}
```

41. Irudia: GoibururikEzObjectOutputStream klasea

```java
public class GoibururikEzObjectInputStream extends ObjectInputStream{

    /* ERAIKITZAILEAK */
    public GoibururikEzObjectInputStream(InputStream is) throws IOException {
        super(is);
    }

    protected GoibururikEzObjectInputStream() throws IOException, SecurityException {
        super();
    }

    /* METODOAK */
    /* fitxategiko kabezera ez irakurtzeko metodoa berridatzi, ezer ez egiteko */
    @Override
    protected void readStreamHeader() throws IOException {
    }
}
```

42. Irudia: GoiburirikEzObjectInputStream klasea

3.9. Erroreen kontrola

Datuak jasotzerako orduan, errore desberdinak agertu daitezke. Erabiltzaileak datu mota egokia ez jartzea, programa datuak jaso edo bueltatzeko kapaza ez izatea…

Errore hauek, ondorengo irudikoen antzerakoak izango dira.

```
Exception in thread "main" java.lang.NumberFormatException: For input string: "d"
        at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
        at java.lang.Integer.parseInt(Integer.java:580)
        at java.lang.Integer.parseInt(Integer.java:615)
        at proiektua_denda.Proiektua_denda.main(Proiektua_denda.java:42)
```

43. Irudia: Adibidea - Errorea

Errore hauek kontrolatzeko, try – cath erabili da. Hauek, errore bat ematen duenean, erabiltzaileak ulertuko ez duen linea gorriak ikusi beharrean, zein izan den arazoa esango dion mezu bat erakutsiko dute.

Hona hemen, erabili diren adibide batzuk:

```
try {
    aukera = Integer.parseInt(br.readLine());
}
catch (NumberFormatException datuOkerrak) {
    System.out.println("Zenbaki bat sartu behar zenuen.");
}
catch (IOException gaizki) {
    System.out.println("Arazoak daude datuak sartzerakoan.");
}
```

44. Irudia: Adibidea - try/catch

Try/catch erabilita, honelako mezuak agertuko dira.

```
 Aukeratu: d
Zenbaki bat sartu behar zenuen.

BUILD SUCCESSFUL (total time: 2 seconds)
```

45. Irudia: Adibidea - try/catch mezuak

Proiektuan erabilitakoak hauek dira:

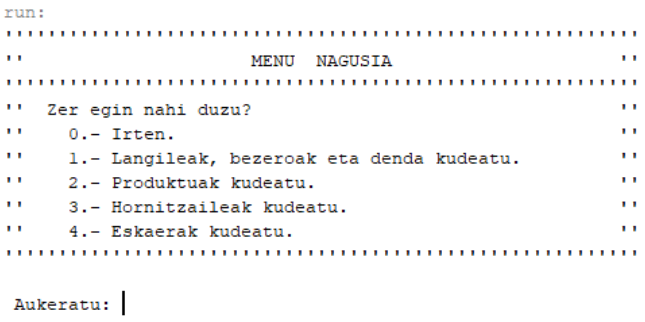
NumberFormatException	Karaktere ez numerikoak sartzean ematen duten erroreak kontrolatzeko.
IOException	Sarrera/irteerako erroreak kontrolatzeko.
ParseException	String bat beste datu mota batera parseatu ezin denean ematen duen errorea kontrolatzeko.

2. Taula: Erroreen kontrola

4. KONTSOLA

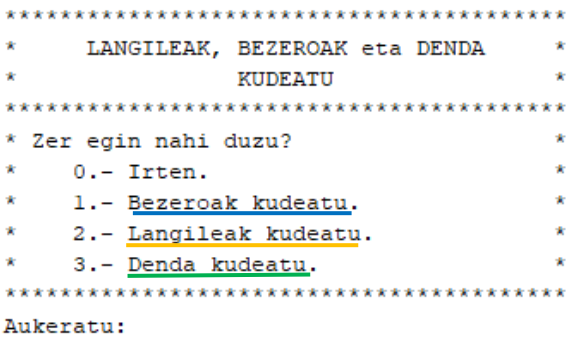
Objektuak sortuko dira eta erabiltzaileak datuak sartuko ditu eta ArrayList-etan gorde, baina, aplikazioa ixten denean, datu horiek desagertu egingo dira.

Menu nagusi bat agertuko da, aukera desberdinak emanez. Aukeratzen den zenbakiaren arabera, beste menu batzuetara bideratuko du.



46. Irudia: Main - Menu nagusia

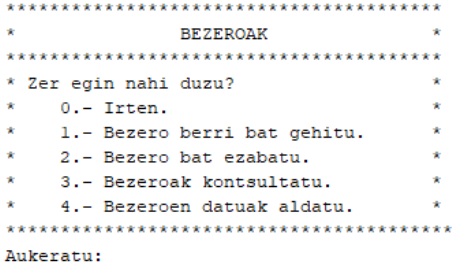
- Menu nagusian **1** zenbakia aukeratzen bada, langileak, bezeroak eta denda kudeatzeko menu bat agertuko da, non aukeratu beharko den zer den nahi dena kudeatu.



47. Irudia: Main - Langileak, bezeroak eta denda kudeatzeko menua

Azpi menu honetako aukerak erakutsiko dira:

- 1 zenbakia aukeratzen bada, **bezeroak** kudeatzeko beste menu bat agertuko da.



0 zenbakia aukeratzen bada, menutik irtengo da.

1 zenbakia aukeratzen bada, bezero berri bat gehituko da. Horretarako, erabiltzaileari eskatuko zaio datuak banan-banan sartzen joateko.

```
Bezero berriaren datuak sartu behar dituzu.

Sartu izena: Oihane
Sartu lehenengo abizena: Axpe
Sartu bigarren abizena: Telleriarte
Sartu NAN zenbakia: 44258962Q
Sartu jaiotze data (uuuu/hh/ee): 1993/05/12
Emakumea edo gizona?: Emakumea
Sartu herria: Bergara
Sartu telefono zenbakia: 666555444
```

Datu guztiak sartzen bukatzean, datuak inprimatuko dira pantailan.

```
Datu hauek dituen bezeroa gorde da.

Eremua: Bezeroa
Kodea: bezeroa#1001
Izen abizenak: Oihane Axpe Telleriarte
NAN: 44258962Q
Jaiotze data: 1993/05/12
Sexua: Emakumea
Herria: Bergara
Tlf: 666555444

Sakatu 'Enter' jarraitzeko...
```

2 zenbakia aukeratzen bada, bezero bat ezabatuko da. Horretarako, erabiltzaileari eskatuko zaio bezeroaren NAN zenbakia sartzeko.

```
Sartu NAN zenbakia: 44348961q
44348961q zenbakidun bezeroa ezabatu da.
```

3 zenbakia aukeratzen bada, erregistratuta dauden bezero guztiak erakutsiko ditu.

```
BEZEROAK:
        Bezero kodea    Izena       Abizenak            NAN zenbakia    Jaiotze data       Sexua         Herria       Telefonoa
        bezeroa#1001    Oihane      Axpe Telleriarte      44258962Q     1993/05/12         Emakumea      Eibar        666555444
```

4 zenbakia aukeratzen bada, bezero baten datuak aldatuko dira. Horretarako, erabiltzaileari eskatuko bezeroaren NAN zenbakia sartzeko, ondoren, aldatzea nahi duen datua aukeratu beharko du, gero aldatzeko.

```
Sartu datuak aldatu nahi dituzun bezeroaren NAN zenbakia: 44258962Q

Zein datu aldatu nahi duzu?
        1.- Izena.
        2.- Lehenengo abizena.
        3.- Bigarren abizena.
        4.- NAN zenbakia.
        5.- Jaiotze data.
        6.- Sexua.
        7.- Herria.
        8.- Telefonoa.
        Beste EDOZEIN ZENBAKI ezer ez aldatzeko.
Aukeratu: 7
Sartu herria: Eibar

Aldatutako datua gorde da.
```

- 2 zenbakia aukeratzen bada, **langileak** kudeatzeko beste menu bat agertuko da.

```
****************************************
*              LANGILEAK               *
****************************************
* Zer egin nahi duzu?                  *
*    0.- Irten.                        *
*    1.- Langile berri bat gehitu.     *
*    2.- Langile bat ezabatu.          *
*    3.- Langileak kontsultatu.        *
*    4.- Langileen datuak aldatu.      *
****************************************
Aukeratu:
```

0 zenbakia aukeratzen bada, menutik irtengo da.

1 zenbakia aukeratzen bada, langile berri bat gehituko da. Horretarako, erabiltzaileari eskatuko zaio datuak banan-banan sartzen joateko.

```
Sartu izena: Miren
Sartu lehenengo abizena: Arregi
Sartu bigarren abizena: Heriz
Sartu NAN zenbakia: 44223355s
Sartutako nan zenbakia ez da egokia.
Sartu NAN zenbakia: 44348960S
Sartu jaiotze data (uuuu/hh/ee): 1990/12/03
Emakumea edo gizona?: Emakumea
Sartu herria: Eibar
Sartu telefono zenbakia: 665874896
Sartu soldata: 1600
Sartu langilearen eremua (saltzailea edo garbitzailea): saltzailea
```

Datu guztiak sartzen bukatzean, datuak inprimatuko dira pantailan.

```
Eremua: Langilea
Kodea: langilea#1001
Lan eremua: saltzailea
Izen abizenak: Miren Arregi Heriz
NAN: 44348960S
Jaiotze data: 1990/12/03
Sexua: Emakumea
Herria: Eibar
Tlf: 665874896
Soldata: 1600,00€

Sakatu 'Enter' jarraitzeko...
```

2 zenbakia aukeratzen bada, langile bat ezabatuko da. Horretarako, erabiltzaileari eskatuko zaio langilearen NAN zenbakia sartzeko.

```
Sartu NAN zenbakia: 44223355S
44223355S zenbakidun langilea ezabatu da.
```

3 zenbakia aukeratzen bada, erregistratuta dauden langile guztiak erakutsiko ditu.

```
LANGILEAK:
    Langile kodea   Izena       Abizenak            NAN zenbakia  Jaiotze data      Sexua       Herria      Telefonoa     Lan-eremua
    langilea#1001   Miren       Arregi Heriz        44348960S     1990/12/03        Emakumea    Eibar       665874896     saltzailea
```

4 zenbakia aukeratzen bada, langile baten datuak aldatuko dira. Horretarako, erabiltzaileari eskatuko langilearen NAN zenbakia sartzeko, ondoren datuak banan-banan sartzen joateko

```
Sartu datuak aldatu nahi dituzun bezeroaren NAN zenbakia: 44348960s

Zein datu aldatu nahi duzu?
        1.- Izena.
        2.- Lehenengo abizena.
        3.- Bigarren abizena.
        4.- NAN zenbakia.
        5.- Jaiotze data.
        6.- Sexua.
        7.- Herria.
        8.- Telefonoa.
        9.- Soladata.
        10.- Eremua (saltzaile/garbitzaile).
        Beste EDOZEIN ZENBAKI ezer ez aldatzeko.
Aukeratu: 3
Sartu bigarren abizena: Eriz
```

- 3 zenbakia aukeratzen bada, **denda** kudeatzeko beste menu bat agertuko da.

```
****************************************
*                DENDA                 *
****************************************
* Zer egin nahi duzu?                  *
*    0.- Irten.                        *
*    1.- Dendaren datuak kontsultatu.  *
*    2.- Dendaren datuak aldatu.       *
****************************************
Aukeratu:
```

0 zenbakia aukeratzen bada, menutik irtengo da.

1 zenbakia aukeratzen bada, dendaren datu guztiak erakutsiko ditu.

```
DENDAREN DATUAK:
        Izena: Denda
        Helbidea: San Juan kalea, 3
        Herria: Eibar
        Posta kodea: 20600
        Email-a: denda@gmai.com
        Telefonoa: 943201258

Sakatu 'Enter' jarraitzeko...
```

2 zenbakia aukeratzen bada, dendaren datuak aldatuko dira. Erabiltzaileak aukeratu beharko du zein datu aldatu nahi duen eta horren arabera, datua sartu.

```
Zer aldatu nahi duzu?
(0) Ezer.
(1) Izena.
(2) Helbidea.
(3) Herria.
(4) Posta kodea.
(5) Email-a.
(6) Telefonoa.
Aukeratu:
```

```
Aukeratu: 2
Sartu helbidea: San Juan kalea, 12

Sakatu 'Enter' jarraitzeko...

Aldaketak egin ondoren, dendaren datuak honako hauek dira.
DENDAREN DATUAK:
        Izena: Denda
        Helbidea: San Juan kalea, 12
        Herria: Eibar
        Posta kodea: 20600
        Email-a: denda@gmai.com
        Telefonoa: 943201258

Sakatu 'Enter' jarraitzeko...
```

- Menu nagusian **2** zenbakia aukeratzen bada, produktuak kudeatzeko menu bat agertuko da.

```
************************************************
*              PRODUKTUAK KUDEATU              *
************************************************
* Zer egin nahi duzu?                          *
*    0.- Irten                                 *
*    1.- Produktuak gehitu.                    *
*    2.- Produktuak ezabatu.                   *
*    3.- Produktuak erakutsi.                  *
*    4.- Produktua dendan dagoen kontsultatu.  *
*    5.- Produktuen inbentarioa.               *
*    6.- Eskatzeko produktuen zerrenda ikusi.  *
*    7.- Produktua saldu.                      *
************************************************

  Aukeratu:
```

48. Irudia: Main - Produktuak kudeatzeko menua

  - 1 zenbakia aukeratzen bada, **produktuak gehitzeko** aukera ematen du eta beste menu bat agertuko da, produktu motaren arabera egiteko.

```
Produktu berri bat gehitu behar duzu.

*************************
    0.- Irten.
    1.- Jertsea.
    2.- Kamiseta.
    3.- Praka.
*************************
Aukeratu:
```

0 zenbakia aukeratzen bada, menutik irtengo da.

1 zenbakia aukeratzen bada, JERTSE berri bat gehituko da. Horretarako, erabiltzaileari eskatuko zaio datuak banan-banan sartzen joateko.

```
Jertse berriaren datuak sartu behar dituzu.
Sartu produktuaren kodea (erreferentzia): 1206596-8223
Sartu marka: Ternua
Sartu prezioa: 44.99
Sartu kolorea: Beltza
Emakumea edo gizona?: Gizona
Sartu Kantitatea: 15
Sartu taila (XS,S,M,L,XL,XXL): M
```

2 zenbakia aukeratzen bada, KAMISETA berri bat gehituko da. Horretarako, erabiltzaileari eskatuko zaio datuak banan-banan sartzen joateko.

```
Kamiseta berriaren datuak sartu behar dituzu.

Sartu produktuaren kodea (erreferentzia): 212103-524
Sartu marka: Neak peak
Sartu prezioa: 9.99
Sartu kolorea: Urdiña
Emakumea edo gizona?: Emakumea
Sartu Kantitatea: 10
Sartu taila (XS,S,M,L,XL,XXL): M
Sartu sasoia (uda, negua...): uda
Datu hauek dituen produktua gorde da.
```

3 zenbakia aukeratzen bada, <u>PRAKA</u> berri bat gehituko da. Horretarako, erabiltzaileari eskatuko zaio datuak banan-banan sartzen joateko.

```
Praka berriaren datuak sartu behar dituzu.
Sartu produktuaren kodea (erreferentzia): BQ3585
Sartu marka: Adidas
Sartu prezioa: 59.99
Sartu kolorea: Beltza
Emakumea edo gizona?: Emakumea
Sartu Kantitatea: 10
Sartu taila (38,40,42,44,46): 40
Sartu luzeera: 80
Sartu mota: Elastikoa
```

- 2 aukeratzen bada, **produktua ezabatzeko** aukera ematen du eta beste menu bat agertuko da, produktu motaren arabera egiteko.

```
Zein produktu ezabatu nahi duzu?
**************************
    0.- Irten.
    1.- Jertsea.
    2.- Kamiseta.
    3.- Praka.
**************************
Aukeratu:
```

0 zenbakia aukeratzen bada, menutik irtengo da.

1 zenbakia aukeratzen bada, <u>JERTSE</u> bat ezabatuko da. Erabiltzaileak, jertsearen kodea sartu beharko du.

```
Sartu ezabatzea nahi duzun jertsearen kodea: 1206596-8223
1206596-8223 erreferentziadun jertsea ondo ezabatu da.
```

2 zenbakia aukeratzen bada, <u>KAMISETA</u> bat ezabatuko da. Erabiltzaileak, kamisetaren kodea sartu beharko du.

```
Sartu ezabatzea nahi duzun kamisetaren kodea: 212103-524
212103-524 erreferentziadun jertsea ondo ezabatu da.
```

3 zenbakia aukeratzen bada, <u>PRAKA</u> bat ezabatuko da. Erabiltzaileak, prakaren kodea sartu beharko du.

```
Sartu ezabatzea nahi duzun kamisetaren kodea: BQ3585
BQ3585 erreferentziadun praka ondo ezabatu da.
```

- 3 zenbakia aukeratzen bada, **produktu guztiak erakusten** ditu, produktu motaren arabera sailkatuta.

```
-----------------------------------------
|        Dendako produktu guztiak       |
-----------------------------------------
KAMISETAK:
        Kodea           Marka           Kolorea         Sexua           Prezioa         Tailak          Sasoia
        CE5205          Adidas          Urdina          Gizona          24.99           M               Uda
        CE5206          Adidas          Granatea        Gizona          24.99           L               Uda
        212103-524      Neak Peak       Berdea          Emakumea        9.99            XS              Uda
        212103-524      Neak Peak       Berdea          Emakumea        9.99            M               Uda
        212103-524      Neak Peak       Berdea          Emakumea        9.99            L               uDA
JERTSEAK:
        Kodea           Marka           Kolorea         Sexua           Prezioa         Tailak
        1206596-8223    Ternua          Beltza          Gizona          44.99           M
        856827-433      Nike            Urdina          Gizona          39.99           L
        CD9620          Adidas          Grisa           Emakumea        66.99           M
        CD9620          Adidas          Grisa           Emakumea        66.99           S

PRAKAK:
        Kodea           Marka           Kolorea         Sexua           Prezioa         Tailak          Luzeera         Mota
        1273283-9937    Ternua          Beltza          Gizona          99.99           38              80              Normala
        1273283-9937    Ternua          Beltza          Gizona          99.99           40              80              Normala
        1273283-9937    Ternua          Beltza          Gizona          99.99           42              80              Normala
        CZ9647          Adidas          Beltza          Emakumea        45.99           38              80              Elastikoa
        CZ9647          Adidas          Beltza          Emakumea        45.99           40              80              Elastikoa


Sakatu 'Enter' jarraitzeko...
```

- 4 zenbakia aukeratzen bada, **produktu zehatz bat dendan dagoen ala ez erakusten** du eta beste menu bat agertuko da, produktu motaren arabera egiteko.

```
Zein produktu mota kontsultatu nahi duzu?
**************************
    0.- Irten.
    1.- Jertsea.
    2.- Kamiseta.
    3.- Praka.
**************************
Aukeratu:
```

0 zenbakia aukeratzen bada, menutik irtengo da.

1 zenbakia aukeratzen bada, <u>JERTSE</u> baten datuak kontsultatuko dira. Horretarako, erabiltzaileari eskatuko zaio jertsearen kodea sartzeko.

```
Sartu produktuaren kodea (erreferentzia): 1206596-8223

Kodea           Kolorea         Taila           Kantitatea
1206596-8223    Beltza          M               15
```

2 zenbakia aukeratzen bada, <u>KAMISETA</u> baten datuak kontsultatuko dira. Horretarako, erabiltzaileari eskatuko zaio kamisetaren kodea sartzeko.

```
Sartu produktuaren kodea (erreferentzia): 2545-524
Kodea   Kolorea Taila   Kantitatea
Ez dago kode hori duen produkturik.
```

3 zenbakia aukeratzen bada, <u>PRAKA</u> baten datuak kontsultatuko dira. Horretarako, erabiltzaileari eskatuko zaio prakaren kodea sartzeko.

```
Sartu produktuaren kodea (erreferentzia): BQ3585
Kodea   Kolorea Taila   Kantitatea
BQ3585  Beltza  40      9
```

- 5 zenbakia aukeratzen bada, **dendako produktuen inbentarioa erakusten** du, produktu motaren arabera sailkatuta.

```
------------------------------------------------------------
|        Dendako produktuen inbentarioa (produktuka)       |
------------------------------------------------------------

KAMISETAK:
        Kodea           Marka       Sexua       Kantitatea
        CE5205          Adidas      Gizona      4
        CE5206          Adidas      Gizona      15
        212103-524      Neak Peak   Emakumea    10
        212103-524      Neak Peak   Emakumea    10
        212103-524      Neak Peak   Emakumea    10

JERTSEAK:
        Kodea           Marka       Sexua       Kantitatea
        1206596-8223    Ternua      Gizona      15
        856827-433      Nike        Gizona      10
        CD9620          Adidas      Emakumea    4
        CD9620          Adidas      Emakumea    12

PRAKAK:
        Kodea           Marka       Sexua       Kantitatea
        1273283-9937    Ternua      Gizona      10
        1273283-9937    Ternua      Gizona      10
        1273283-9937    Ternua      Gizona      10
        CZ9647          Adidas      Emakumea    5
        CZ9647          Adidas      Emakumea    15
```

- 6 zenbakia aukeratzen bada, **dendako produktuen artean, stock-ean 5 baino gutxiago daudenak erakusten** ditu, produktu motaren arabera sailkatuta.

```
-------------------------------------------
|          Eskatzeko produktuak           |
-------------------------------------------

KAMISETAK:
        Kodea           Marka       Sexua       Kantitatea
        CE5205          Adidas      Gizona      4

JERTSEAK:
        Kodea           Marka       Sexua       Kantitatea
        CD9620          Adidas      Emakumea    4

PRAKAK:
        Kodea           Marka       Sexua       Kantitatea
        Ez dago eskatzeko produkturik.
```

- 7 zenbakia aukeratzen bada, **produktuak saltzeko** aukera ematen du eta beste menu bat agertuko da, produktu motaren arabera egiteko.

```
Produktu bat saldu behar duzu.

**************************
    0.- Irten.
    1.- Jertsea.
    2.- Kamiseta.
    3.- Praka.
**************************
Aukeratu:
```

0 zenbakia aukeratzen bada, menutik irtengo da.

1 zenbakia aukeratzen bada, **JERTSE** bat salduko da. Horretarako, erabiltzaileari eskatuko zaio jertsearen kodea, taila eta erosi nahi duen kantitatea sartzeko.

```
Sartu saldu behar duzun produktuaren kodea: 1206596-8223
Sartu taila: M
Sartu kantitatea: 1
----------------------------------------
|         Produktuaren salmenta         |
----------------------------------------
        Kodea   -       Prezioa

        1206596-8223    -       44.99   (x1)
  ------------------------------------
        ORDAINTZEKOA
        GUZTIRA:                44.99€

Sakatu 'Enter' jarraitzeko...
```

2 zenbakia aukeratzen bada, **KAMISETA** bat salduko da. Horretarako, erabiltzaileari eskatuko zaio kamisetaren kodea, taila eta erosi nahi duen kantitatea sartzeko.

```
Sartu saldu behar duzun produktuaren kodea: 212103-524
Sartu taila (XS,S,M,L,XL,XXL): M
Sartu kantitatea: 2
----------------------------------------
|         Produktuaren salmenta         |
----------------------------------------
        Kodea   -       Prezioa

        212103-524      -       9.99    (x2)
  ------------------------------------
        ORDAINTZEKOA
        GUZTIRA:                19.98€

Sakatu 'Enter' jarraitzeko...
```

3 zenbakia aukeratzen bada, **PRAKA** bat salduko da. Horretarako, erabiltzaileari eskatuko zaio prakaren kodea, taila eta erosi nahi duen kantitatea sartzeko.

```
Sartu saldu behar duzun produktuaren kodea: BQ3585
Sartu taila (38,40,42,44,46): 40
Sartu kantitatea: 1
----------------------------------------
|         Produktuaren salmenta         |
----------------------------------------
        Kodea   -       Prezioa

        BQ3585  -       59.99   (x1)
  ------------------------------------
        ORDAINTZEKOA
        GUZTIRA:                59.99€

Sakatu 'Enter' jarraitzeko...
```

- Menu nagusian **3** zenbakia aukeratzen bada, hornitzaileak kudeatzeko menu bat agertuko da.

```
************************************************
*             HORNITZAILEA KUDEATU             *
************************************************
* Zer egin nahi duzu?                          *
*    0.- Irten                                 *
*    1.- Hornitzailea gehitu.                  *
*    2.- Hornitzailea ezabatu.                 *
*    3.- Hornitzaile guztiak erakutsi.         *
************************************************
```

49. Irudia: Main - Hornitzaileak kudeatzeko menua

  - 1 zenbakia aukeratzen bada, **hornitzaile berri bat gehitzeko** aukera ematen du eta erabiltzaileak, hornitzailearen datuak sartu beharko ditu.

```
Hornitzaile berri bat gehitu behar duzu.

Hornitzaile berriaren datuak sartu behar dituzu.
Sartu hornitzailearen kodea: 112233
Sartu izena: Adidas
Sartu herria: Zaragoza
Sartu telefonoa: 976710100
Sartu email-a: adidas@adidas.com

Hornitzailearen datuak:
        Kodea: 112233
        Izena: Adidas
        Herria: Zaragoza
        Telefonoa: 976710100
        Email-a: adidas@adidas.com

Sakatu 'Enter' jarraitzeko...
```

  - 2 zenbakia aukeratzen bada, **hornitzaile bat ezabatzeko** aukera ematen du eta erabiltzaileak, hornitzailearen kodea sartu beharko du, honen datuak ezabatu ahal izateko.

```
Sartu hornitzailearen kodea: 112233
Hornitzailea ondo ezabatu da.
```

  - 3 zenbakia aukeratzen bada, **erregistratuta dauden hornitzaileen informazioa** erakutsiko du.

```
HORNITZAILEAK:
        Kodea                   Izena          Herria         Telefonoa          Email-a
        hornitzailea#1001       Adidas         Madrid         645123654          support@adidas.com
```

- Menu nagusian **4** zenbakia aukeratzen bada, eskaerak kudeatzeko menu bat agertuko da.

```
************************************************
*                ESKAERA KUDEATU               *
************************************************
* Zer egin nahi duzu?                          *
*    0.- Irten                                 *
*    1.- Eskaera berri bat gehitu.             *
*    2.- Eskaera guztiak erakutsi.             *
************************************************

  Aukeratu:
```

50. Irudia: Main - Hornitzaileak kudeatzeko menua

  - 1 zenbakia aukeratzen bada, **eskaera berri bat gehitzeko** aukera ematen du eta erabiltzaileak, hornitzailearen datuak sartu beharko ditu.

```
Eskaera berri bat gehitu behar duzu.

Eskaera berriaren datuak sartu behar dituzu.

Sartu Hornitzailea: Ternua
Sartu kopurua: 10


Datu hauek dituen eskaera gorde da.
Eskaera zenbakia: eskaera#1007
Hornitzailea: Ternua
Data: 2018/04/16
Kopurua: 10

Sakatu 'Enter' jarraitzeko...
```

  - 2 zenbakia aukeratzen bada, **erregistratuta dauden eskaeren informazioa** erakutsiko du.

```
ESKAERAK:
        Eskaera zenb        Hornitzailea        Kopurua        Data
        eskaera#1002        Adidas              12             03/04/2018
        eskaera#1003        Nike                12             03/04/2018
        eskaera#1001        Trango              5              03/04/2018
        eskaera#1004        Adidas              10             03/04/2018
        eskaera#1005        Adidas              10             009/10/09
        eskaera#1006        Adidas              15             2018/04/04
```