**DOKUMENTAATIO**

*Työ V4: Toteuta luokassa Varasto metodit, jolla voidaan poistaa tuote sekä metodi, joka palauttaa tuotteen, jota varastossa on eniten. Tee luokkaan Varasto myös metodi, jolla voidaan muuttaa annetun tuotteen hintaa.*

**1. Ratkaisuperiaate**

Ongelmanratkaisua varten kirjoitetaan luokat Varasto ja Varastontuote. Niiden lisäksi kirjoitetaan testiluokka VarastoTesti, jolla kokeillaan Varaston ja VarastonTuotteen toimivuutta.

Luokka Varasto mallintaa varastoa, jossa säilytetään tuotteita. Varaston tietokenttiä ovat nimi (tyyppiä String) , osoite (tyyppiä String) ja tuotteet (tyyppiä LinkedList).

Konstruktorilla luodaan uusi varasto, kun sille on annettu parametreina varaston nimi ja osoite. Samalla luodaan myös tyhjä tuotelista, johon lisätään myöhemmin varastossa säilytettävät VarastonTuote-typpiset tuotteet.

Havainnointimetodeilla voidaan hakea varaston nimi, osoite ja tuotteet ja palauttaa tai tulostaa ne.

Muutosmetodeilla voidaan lisätä ja poistaa tuotelistasta tuotteita, muuttaa tuotteiden hintaa sekä muuttaa varaston nimeä tai osoitetta

Luokan Varastontuote tietokenttiä ovat nimi (tyyppiä String), id (tyyppiä int), hinta (tyyppiä double) ja määrä(tyyppiä int)

Konstruktoreilla voidaan luoda VarastonTuote-olio, josta voidaan jättää puuttumaan määrä (jolloin sen arvoksi asetetaan nolla)

Havainnointimetodeilla voidaan hakea tuotteen (tai pikemminkin tuote-erän) nimi, id, hinta ja määrä. Edellämainitut arvot voidan myös tulostaa.

Muutosmetodeilla voidaan lisätä tai vähentää tuotteen määrää tai muuttaa se tiettyyn arvoon.

Luokka VarastoTesti testaa luokkien Varasto ja VarastoTesti metodien toimivuutta. Luokassa käytetään kaikkia edellämainituissa luokissa määriteltyjä metodeja ja muutokset tulostetaan suorituksen jälkeen näytölle, jolloin testaaja voi tehdä johtopäätöksiä luokkien toimivuudesta. Ensin testataan luokan VarastonTuote metodeja ja sen jälkeen luokan Varasto.

**2. Ohjelman ja sen osien kuvaaminen**

**2.1 Luokka Varasto**

kuvaa varastoa, joka sisältää VarastonTuote-tyyppisiä olioita

**Tietokentät**

/\*\*

\* Varaston nimi.

\*/

private String nimi;

/\*\*

\* Varaston osoite.

\*/

private String osoite;

/\*\*

\* Varaston tuotteet dynaamisena listana.

\*/

private LinkedList<VarastonTuote> tuotteet;

**Konstruktori**

/\*\*

\* Luo varasto-olion, jonka nimi on n ja osoite on o. Luo tyhjän tuotelistan.

\*/

public Varasto (String n, String o)

**Havainnointimetodit**

/\*\*

\* Palauttaa varaston osoitteen merkkijonona.

\*/

public String getOsoite()

/\*\*

\* Palauttaa varaston nimen merkkijonona.

\*/

public String getNimi()

/\*\*

\* Palauttaa sen VarastonTuote-olion, jonka id on sama kuin parametrinä annettu id.

\* AE: id:tä vastaava tuote pitää olla olemassa

\*/

public VarastonTuote getTuote(int id)

/\*\*

\* Palauttaa tuotteen jonka määrä on varastossa suurin.

\* Jos suurilukuisimpia tuotteita on useita, metodi palauttaa listassa ensimmäisenä olevan

\* AE: varasto ei ole tyhjä

\*/

public VarastonTuote getEniten()

/\*\*

\* Tulostaa varasto-olion kaikkien tietokenttien arvot.

\*/

public void tulosta()

**Muutosmetodit**

/\*\*

\* Muuttaa varaston osoitteen parametrinä annetuksi merkkijonoksi.

\*/

public void setOsoite(String o)

/\*\*

\* Muuttaa varaston nimen parametrinä annetuksi merkkijonoksi.

\*/

public void setNimi(String n)

/\*\*

\* Muuttaa varastossa olevan tuotteen hintaa.

\*/

public void setTuotteenHinta (int id, double hinta)

/\*\*

\* Lisää varastoon valmiiksi luodun tuotteen.

\*/

public void lisaaTuote(VarastonTuote t)

/\*\*

\* Poistaa tuotelistasta parametrinä annettua id:tä vastaavan tuotteen.

\*/

public void poistaTuote (int id)

/\*\*

\* Vähentää/lisää varastossa olevan tuotteen määrää. Positiivinen luku lisää

\* tuotteen määrää ja negatiivinen vähentää sitä.

\* AE: tuotteen määräksi ei tule negatiivinen luku

\*/

public void muutaTuotteenMaaraa (int tunniste, int m)

**2.2 Luokka VarastonTuote**

Kuvaa Varasto-tyyppisessä varastossa säilytettäviä tuotteita

**Tietokentät**

/\*\*

\* Tuotteen nimi.

\*/

private String nimi;

/\*\*

\* Tuotteen id.

\*/

private int id;

/\*\*

\* Tuotteen hinta.

\*/

private double hinta;

/\*\*

\* Tuotteen määrä varastossa.

\*/

private int maara;

**Konstruktorit**

/\*\*

\* Luo VarastonTuote-olion, jonka nimi on n, id on tunniste, hinta on h ja määrä on m.

\*/

public VarastonTuote (String n, int tunniste, double h, int m)

/\*\*

\* Luo VarastonTuote-olion, jonka nimi on n, id on tunniste ja hinta on h. Määräksi asetetaan 0.

\*/

public VarastonTuote (String n, int tunniste, double h)

**Havainnointimetodit**

/\*\*

\* Palauttaa tuotteen nimen.

\*/

public String getNimi()

/\*\*

\* Palauttaa tuotteen id:n.

\*/

public int getId()

/\*\*

\* Palauttaa tuotteen hinnan.

\*/

public double getHinta()

/\*\*

\* Palauttaa tuotteen määrän.

\*/

public int getMaara()

/\*\*

\* Tulostaa tuotteen nimen, tunnisteen, hinnan ja määrän.

\*/

public void tulosta()

/\*\*

\* Palauttaa tuotteen tiedot merkkijonona.

\*/

public String toString()

**Muutosmetodit**

/\*\*

\* Muuttaa tuotteen nimen paramterinä annetuksi merkkijonoksi.

\*/

public void setNimi(String n)

/\*\*

\* Muuttaa tuotteen id:n parametrinä annetuksi id:ksi.

\*/

public void setId(int tunniste)

/\*\*

\* Muuttaa tuotteen hinnan parametrinä annetuksi hinnaksi.

\*/

public void setHinta(double h)

/\*\*

\* Muuttaa tuotteen määräksi parametrina annetun arvon.

\*/

public void setMaara(int m)

/\*\*

\* Vähentää/lisää varastossa olevan tuotteen määrää. Positiivinen luku lisää

\* tuotteen määrää annetun luvun verran ja negatiivinen luku vähentää sitä.

\* AE: tuotteen määräksi ei tule negatiivinen luku

\*/

public void muutaMaaraa (int lkm)

**2.3 Luokka VarastoTesti**

Luokkien Varasto ja VarastonTute testaamiseen tehty luokka.

**Staattiset metodit**

/\*\*

\* Testausluokan päämetodi.

\*/

public static void main (String[]args)

/\*\*

\* Testaa luokan Varasto toimivuutta.

\*/

public static void testaaVarasto()

/\*\*

\* Testaa luokan VarastonTuote toimivuutta.

\*/

public static void testaaTuote()

**3. Testausjärjestelyt**

Testaus suoritetaan siten, että ensin testataan luokan VarastonTuote toimivuus ja sitten luokan Varasto. Luokka sisältää myös tyhjien rivien tulostuksia, jotka tekevät siitä käyttäjäystävällisemmän.

**3.1 Luokan VarastonTuote testaus**

1. Luodaan VarastonTuote-tyyppisiä oliota

2. Tulostetaan niiden tietokenttien arvot käyttäen kaikkia havainnointimetodeja

3. Muutetaan olioiden tietokenttiä käyttäen kaikkia muutosmetodeja ja tulostetaan muuttuneet arvot

Edellä olevien vaiheiden jälkeen ohjelma tulostaa:

Vasaroita

1

20

10.0

Vasaroita/1/20/10.0

Nimi: Nauloja

ID: 2

Hinta: 5.0

Määrä: 0

Nimi: Leka

ID: 10

Hinta: 15.0

Määrä: 50

Nimi: Nauloja

ID: 2

Hinta: 5.0

Määrä: 10

Nimi: Ruuvimeisseleitä

ID: 3

Hinta: 1.0

Määrä: 1

**3.2 Luokan Varasto testaus**

1. Luodaan VarastonTuote-tyyppisiä oliota

2. Luodaan Varasto-tyyppinen olio ja lisätään edellisessä kohdassa luodut tuotteet varastoon

3. Tulostetaan varaston tietokenttien arvot käyttäen kaikkia havainnointimetodeja

4. Muutetaan varaston ja siinä olevien tuotteiden tietokenttiä käyttäen kaikkia luokasta Varasto löytyviä muutosmetodeja ja tulostetaan muuttuneet arvot

Edellä olevien vaiheiden jälkeen ohjelma tulostaa:

Varastokatu 3

Matin varasto

Vasaroita/1/20/10.0

Vasaroita/1/20/10.0

Nimi: Matin varasto

Sijainti: Varastokatu 3

Sisältö:

Vasaroita/1/20/10.0

Nauloja/2/0/5.0

Ruuvimeisseleitä/3/1/1.0

Ruuvimeisseleitä/3/21/1.0

Nimi: Raimon varasto

Sijainti: Varastotie 14

Sisältö:

Vasaroita/1/10/15.0

Ruuvimeisseleitä/3/21/1.0

21.4.2013

Santeri Saariokari

Paimion Lukio