

TEKIJÄT:

[1]

Saliavustaja

Tiedosto

Tilausnumero	Tilauksen kokonaishinta ALV0	Tapahtuman tila
1	57,00 €	Vahvistettu
2	34,00 €	Vahvistettu
3	34,00 €	Vahvistettu
4	137,70 €	Vahvistettu
5	0,00 €	Vahvistettu

LISTVIEW

Uusi tilaus

Valitse pöytä

Pöytä 5, paikkoja 10, VAPAA

COMBOBOX

CHECKBOX

☒ Bonusasiakas, 15% ale

Aterian nimi	Hinta
Lämminsavustettua hanhenrintaa ja fen...	19,38 €
Hirvipastramia ja viinäisen uppomuna	19,38 €
Siikaa ja jokirapua	19,38 €
Härkää ja lohta	32,49 €
Paahdettua nieriää ja lime-voikastiketta	32,49 €
Kaldoaivin poronvasaa ja rakuunakasti...	38,76 €
Munakoisoa, paprikaa ja pinjansiementä	30,78 €
Raparperia ja minttujäätelöä 12	13,68 €
Tymipossetti ja luomusuklaata	13,68 €
Valikoima kotimaisia juustoja ja kuusen...	14,82 €

Lisaa >>

<< Poista

Aterian nimi	Hinta	Määrä
Paahdettua nieriää...	32,49 €	2
Raparperia ja mintt...	13,68 €	1
*		

GRIDVIEW

Veroton hinta

Veron osuus

Tilauksen kokonaishinta

58,65 €

8,21 €

66,86 €

Kertyviä etupisteitä

1,3

Peru tilaus

Vahvista tilaus

TEKIJÄT:

[1]

Käyttöliittymä

Käyttöliittymän suunnittelu tapahtuu annettujen kuvien perusteella. Käyttöliittymän komponenttien pitää toteuttaa toiminnollisuudet, jotka on kuvattu seuraavassa kappaleessa. Käyttöliittymän ulkonäkö saa poiketa annetuista kuvista mutta sen tekemät toimenpiteet täytyy olla samanlaisia.

Pääikkunan komponentit:

1. Tiedosto -menu
 - a. Käyttäjä voi aukaista tilauksen vastaanotto-ikkunan tai lopettaa ohjelman suorittamisen.
2. Tilaukset-lista
 - a. Listassa on määritetty sarakkeet tilauksen perustiedoille, josta voi nähdä luotujen tilausrivien määrän. (Tässä tarvitaan tiedostopohjaista tietokannan toteutusta)

Tilauksen vastaanotto -ikkunan komponentit:

1. Valitse pöytä-pudotusvalikko.
 - a. Pudotusvalikosta käyttäjä voi valita pöydän ja nähdä tietoja pöydän tilasta.
2. Bonusasiakas-ruutu.
 - a. Ruudun valitsemalla päälle ja pois määritetään, onko asiakas bonusasiakas. Jos asiakas on bonusasiakas, se vaikuttaa ohjelman toiminnollisuuksiin, jotka on kuvattu luokkien ja metodien kuvausten yhteydessä.
3. Ateria-lista.
 - a. Listassa on määritetty sarakkeet aterian tiedoille, jotka käyttäjälle esitetään: aterian nimi ja hinta.
4. Valitut ateriat-lista.
 - a. Käyttöliittymän toiseen listaan koetaan ateriat, jotka on tilaukseen lisätty. Listassa on sarakkeet aterian nimelle, hinnalle ja määrälle. Käyttäjä voi muuttaa aterioiden määrää tämän listan sarakkeesta.
5. Lisää/poista -napit
 - a. Napit ovat tarkoitettu aterioiden lisäämiseen listasta toiseen. Tapahtuman voi toteuttaa parhaalla näkemällä tavallaan mutta esimerkeissä on käytetty nappeja.
6. Veroton hinta -kenttä
 - a. Ohjelman käyttöliittymässä täytyy näkyä tilauksen veroton kokonaishinta, joka päivittyy sen mukaan kun tilausta muutetaan.
7. Veron osuus -kenttä
 - a. Kenttään lasketaan veron osuus kokonaishinnasta, joka päivittyy sen mukaan kun tilausta muutetaan.
8. Tilauksen kokonaishinta -kenttä
 - a. Kenttään lasketaan tilauksen verollinen kokonaishinta, joka päivittyy sen mukaan kun tilausta muutetaan.
9. Kertyviä etupisteitä -kenttä (bonusasiakas)
 - a. Jos valittuna on bonusasiakas, niin käyttöliittymään paljastetaan kenttä, joka kertoo kertyvien etupisteiden määrän. Etupisteet lasketaan jäljempänä kuvatulla tavalla siitä vastaavan luokan avulla.
10. Peru tilaus -nappi
 - a. Ohjelman ikkuna suljetaan ja tilausta ei suoriteta.
11. Vahvista tilaus -nappi
 - a. Ohjelma vahvistaa tilauksen ja tarkistaa tarvittavat tiedot tilauksesta. Onnistuneen tilauksen vahvistamisen jälkeen siitä ilmoitetaan viestillä käyttäjälle ja ikkuna suljetaan, jolloin ohjelma palaa pääikkunaan. Virheen sattuessa, viestittä lukee virheen syy ja käyttäjä voi jatkaa tilauksen korjaamista sulkematta ikkunaa.

TEKIJÄT:

[2]

```
/// <summary>
/// TilauksenVastaanotto.cs
///
/// Luokan tarkoitus on hoitaa tilauksen käsittely ja tarkistaa käyttäjän
/// antaman syötteen. Luokka ei toteuta muita toimintoja kuin tilauksen
/// vastaanottoon liittyvät toimenpiteet. Tilauksen vastaanotossa tapahtuvat
/// vaiheet on kuvattu metodin kuvauksessa.
///
/// Luokalla on kolme ominaisuutta:
///
/// 1. TilausDb tilausDb
/// 2. PoytaDb poytaDb
/// 3. BonusAsiakasDb asiakasDb
/// </summary>
```

```
/// <summary>
/// TilauksenVastaanotto(TilausDb tilausDb, PoytaDb poytaDb)
///
/// Luokalla on parametrillinen konstruktori, joka ottaa vastaan kaksi
/// tietokantaa käsittelevää oliota. Tietokantaa käsittelevät oliot
/// ovat:
///
/// 1. TilausDb
/// 2. PoytaDb
/// </summary>
/// <param name="tilausDb"></param>
/// <param name="poytaDb"></param>
```

TEKIJÄT:

```
/// <summary>
/// TilauksenVastaanotto(TilausDb tilausDb, PoytaDb poytaDb, BonusAsiakasDb
asiakasDb)
///
/// Luokalla on parametrillinen konstruktori, joka ottaa vastaan kolme
/// tietokantaa käsittelevää oliota. Tietokantaa käsittelevät oliot
/// ovat:
///
/// 1. TilausDb
/// 2. PoytaDb
/// 3. BonusAsiakasDb
/// </summary>
/// <param name="tilausDb"></param>
/// <param name="poytaDb"></param>
/// <param name="asiakasDb"></param>

/// <summary>
/// VastaanotaTilaus(Tilaus tilaus)
///
/// Metodi suorittaa toimenpiteitä, joita tehdään ennen ja jälkeen
/// tilauksen vahvistamisen. VastaanotaTilaus -metodi koostaa ohjelman
/// muiden olioiden toimintoja yhteen. Suoritettavat toimenpiteet ovat
/// seuraavat:
///
/// 1. Metodi saa tilauksen tiedot sille annetusta parametrasta.
///
/// 2. Tarkistetaan tiedot virhesyötteiden varalta. Virhetilanteen
/// sattuesssa tilausta ei vahvisteta ja siitä luodaan virheilmoitus.
/// Virheen sattuesssa se siepataan virheenkäsittelyllä ja annetaan
/// metodia kutsuttavan luokan hoitaa sen käsittely.
///
/// 2.1 VIRHE: Tilausrivien määrä on 0 tai alle.
/// 2.2 VIRHE: Tilaukseen ei ole määritetty asiakasta.
/// 2.3 VIRHE: Tilaukseen ei ole määritetty pöytää.
/// 2.4 VIRHE: Valittu pöytä on jo varattu.
/// 2.5 VIRHE: Tilaus epäonnistuu muusta kuin yllämainituista syistä.
///
/// 3. Vahvistetaan tilaus.
/// 4. Luodaan uusi tilaus tietokantaan.
/// 5. Merkitään pöytä varatuksi.
/// 6. Tarkistetaan onko asiakas bonusasiakas
/// 6.1 Jos on bonusasiakas, kerrytetään etupisteitä ja
/// tallennetaan bonusasiakkaan tiedot tietokantaan.
/// </summary>
/// <param name="tilaus"></param>
```

TEKIJÄT:

[3]

```
/// <summary>
/// Tilaus.cs
///
/// Luokka käsittelee Tilaus-olion tapahtumia ja on yksi
/// ohjelman toimintoja koostavista luokista. Se tarvitsee
/// toimiakseen ohjelman muita luokkia. Luokan toiminnot
/// on tarkemmin kuvattu sen metodien kuvauksissa.
///
/// Luokalla on seuraavia ominaisuuksia:
///
/// 1. int Tilausnumero
/// 2. List<Tilausrivi> Tilausrivit
/// 3. Tapahtumantila TapahtumanTila
/// 4. DateTime Pvm
/// 5. Poyta Poyta
/// 6. Asiakas Asiakas
/// </summary>

/// <summary>
/// Tilaus()
///
/// Parametriton rakentaja. Rakentaja alustaa Tilaus-tyyppisen
/// olion alkuarvoilla, jotka ovat esitelty seuraavaksi.
///
/// Tilausnumero = 0;
/// Tilausrivit = new List<Tilausrivi>();
/// Poyta = null;
/// Asiakas = null;
/// TapahtumanTila = Tapahtumantila.Vahvistamaton;
/// Pvm = DateTime.Now;
/// </summary>

/// <summary>
/// Tilaus(int tilausnumero, List<Tilausrivi> tilausrivit,
///      Poyta poyta, Asiakas asiakas,
///      Tapahtumantila tapahtumanTila, DateTime pvm)
///
/// Parametrillinen konstruktori luo Tilaus-tyyppisen olion
/// sille annetuista alkuarvoista. Parametrit ovat samoja kuin
/// luokan ominaisuudet.
/// </summary>
/// <param name="tilausnumero"></param>
/// <param name="tilausrivit"></param>
/// <param name="poyta"></param>
/// <param name="asiakas"></param>
/// <param name="tapahtumanTila"></param>
/// <param name="pvm"></param>

/// <summary>
/// void LisaaAteria(Ateria ateria, int maara)
///
/// Metodi lisää tilaukseen tilausrivin, joka sisältää
/// parametrina annetun aterian ja määrän.
/// </summary>
/// <param name="ateria"></param>
/// <param name="maara"></param>
```

TEKIJÄT:

```
/// <summary>
/// void PoistaAteria(Ateria ateria)
///
/// Metodi poistaa tilauksen tilausrivistä parametrina
/// annetun aterian. Aterialla on tunnus, jota verrataan
/// tilausriveissä oleviin aterioihin.
/// </summary>
/// <param name="ateria"></param>

/// <summary>
/// void VaihdaAteriaMaara(Ateria ateria, int maara)
///
/// Metodi muuttaa aterian määrää tilauksessa. Tilausrivistä
/// etsitään aterian tunnuksella oikea rivi ja sen määrää
/// muutetaan.
/// </summary>
/// <param name="ateria"></param>
/// <param name="maara"></param>

/// <summary>
/// void VahvistaTilaus()
///
/// Metodi muuttaa Tilaus-olion tilaa siten, että merkitsee
/// tilauksen vahvistetuksi ja asettaa päivämääräksi sen
/// hetkisen ajanhetken.
/// </summary>

/// <summary>
/// bool OnkoVahvistettu()
///
/// Metodi tarkistaa, onko tapahtuman tila vahvistettu.
/// Jos on niin palauttaa true, muussa tapauksessa
/// palautetaan arvo false.
/// </summary>
/// <returns></returns>

/// <summary>
/// double LaskeVerollinenKokonaishinta()
///
/// Metodi laskee ja palauttaa tilauksen verollisen kokonaishinnan.
/// Kokonaishinta lasketaan siten, että tilausrivit käydään läpi
/// ja niiden summat ateriodien määrän mukaan lasketaan kaikki yhteen.
///
/// Jos Asiakas-tyyppinen olio on määritetty tilaukseen niin lasketaan
/// asiakkaalle hinta. Hinta voi vaihdella, jos kyseessä on bonusasiakas.
///
/// Lopuksi metodi palauttaa verollisen kokonaishinnan.
/// </summary>
/// <returns></returns>

/// <summary>
/// double LaskeVerotonKokonaishinta()
///
/// Metodi laskee ja palauttaa tilauksen verottoman kokonaishinnan.
/// Kokonaishinta lasketaan siten, että tilausrivit käydään läpi
/// ja niiden summat ateriodien määrän mukaan lasketaan kaikki yhteen.
///
/// Jos Asiakas-tyyppinen olio on määritetty tilaukseen niin lasketaan
/// asiakkaalle hinta. Hinta voi vaihdella, jos kyseessä on bonusasiakas.
///
/// Lopuksi metodi palauttaa verottoman kokonaishinnan.
/// </summary>
/// <returns></returns>
```

TEKIJÄT:

[3]

```
/// <summary>
/// InMemoryTilausDb.cs
///
/// Luokka käsittelee tietokantayhteyksiä Tilaus-luokan
/// olioiden osalta.
///
/// HUOMAA ETTÄ KYSEESSÄ ON MUISTINVARAINEN TIETOKANTATOTEUTUS!
/// TOTEUTUSTAPAAN VOIT KYSYÄ NEUVOA.
///
/// Luokalla on seuraavia ominaisuuksia:
///
/// 1. Hashtable tilaukset
/// 2. static int seuraavaId = 1
/// </summary>

/// <summary>
/// void Uusi(Tilaus tilaus)
///
/// Metodi tallentaa tilauksen tietokantaan (Hashtable).
/// Parametrina annetaan koko tilaus -olio, joka sisältää tiedot
/// tilauksesta.
/// </summary>
/// <param name="tilaus"></param>

/// <summary>
/// List<Tilaus> HaeKaikki()
///
/// Metodi palauttaa kaikki tietokannan (hashtable) sisällön
/// listana, joka sisältää Tilaus-olioita.
/// </summary>
/// <returns></returns>
```

TEKIJÄT:

[3]

```
/// <summary>
/// FileSystemTilausDb.cs
///
/// Luokka käsittelee tietokantayhteyksiä Tilaus-luokan
/// olioiden osalta.
///
/// HUOMAA ETTÄ KYSEESSÄ ON TIEDOSTOPOHJAINEN TIETOKANTATOTEUTUS!
/// TOTEUTUSTAPAAN VOIT KYSYÄ NEUVOA, KOSKA TALLENNETTAVIEN
/// LUOKKIEN TÄYTYY OLLA SERIALISOITAVIA.
///
/// Luokalla on seuraavia ominaisuuksia:
///
/// 1. Hashtable tilaukset
/// 2. static int seuraavaId = 1
/// 3. string tiedostonPolku = "";
/// 4. BinaryFormatter binaryFormatter = null;
/// </summary>

/// <summary>
/// FileSystemTilausDb(string tiedostonPolku)
///
/// Parametrillinen rakentaja, jolle parametrina annetaan
/// tiedostopolku. Tiedostopolkua käytetään tallennus sijaintina
/// tietokannalle. Esimerkiksi tiedostopolku voi olla muotoa:
///
/// C:\Temp\tietokanta.dat
///
/// Rakentajassa on alustettava myös binaryFormatter -ominaisuus.
/// Tietojenkäsittelyyn tietokanta käyttää Hastablea -muistinvaraisen
/// tietokannan tapaan.
/// </summary>
/// <param name="tiedostonPolku"></param>

/// <summary>
/// void Uusi(Tilaus tilaus)
///
/// Metodi tallentaa tilauksen tietokantaan (Hashtable) ja
/// tietokannan sisältö tallennetaan tietojärjestelmää.
/// Tallennus sijaintina käytetään olion luonnin yhteydessä
/// annettua tiedostopolkua.
///
/// Parametrina annetaan koko tilaus -olio, joka sisältää tiedot
/// tilauksesta.
/// </summary>
/// <param name="tilaus"></param>

/// <summary>
/// Tilaus Hae(int tilausnumero)
///
/// TÄTÄ METODIA EI OLE KÄYTETTY MISSÄÄN
/// </summary>
/// <param name="tilausnumero"></param>
/// <returns></returns>

/// <summary>
/// List<Tilaus> HaeKaikki()
///
/// Metodi palauttaa kaikki tietokannan (hashtable) sisällön
/// listana, joka sisältää Tilaus-olioita.
///
/// Ennen tietojen palauttamista, on tiedot luettava tiedostojärjestelmästä
/// ja luotava Tilaus-olioiksi. Tämän jälkeen oliot voidaan palauttaa
/// listana.
/// </summary>
/// <returns></returns>
```


TEKIJÄT:

[4]

```
/// <summary>
/// Tilausrivi.cs
///
/// Luokka käsittelee tilausrivin tietoja. Luokan
/// toiminnot on kuvattu tarkemmin sen metodien
/// kuvauksissa. Luokalla on seuraavia ominaisuuksia:
///
/// 1. Ateria Ateria
/// 2. int Maara
/// </summary>

/// <summary>
/// Tilausrivi(Ateria ateria, int maara)
///
/// Parametrillinen konstruktori, joka luo Tilausrivi-
/// olion olemassa olevista tiedoista. Parametreina käytetään
/// luokassa esiteltyjä ominaisuuksia.
/// </summary>
/// <param name="ateria"></param>
/// <param name="maara"></param>

/// <summary>
/// void VaihdaMaara(int maara)
///
/// Metodi vaihtaa olion tilaa ja asettaa määräksi
/// sen mikä metodille annetaan parametrina.
/// </summary>
/// <param name="maara"></param>
```

TEKIJÄT:

[4]

```
/// <summary>
/// Ateria.cs
///
/// Luokka käsittelee aterioiden tietoja. Luokan
/// toiminnot on kuvattu tarkemmin sen metodien
/// kuvauksissa. Aterialla on seuraavia ominaisuuksia:
///
/// 1. int Id
/// 2. string Nimi
/// 3. double VerotonHinta
/// 4. readonly double ALV (Miksi tämä ei ole const -tyyppinen vakio?)
/// </summary>

/// <summary>
/// Ateria(int id, string nimi, double verotonHinta, double alv)
///
/// Parametrillinen konstruktori, joka rakentaa olemassa olevista
/// tiedoista Ateria-olion. Parametrit ovat samat kuin luokan
/// esitellyt ominaisuudet.
/// </summary>
/// <param name="id"></param>
/// <param name="nimi"></param>
/// <param name="verotonHinta"></param>
/// <param name="alv"></param>

/// <summary>
/// LaskeVerollinenHinta()
///
/// Metodi palauttaa aterian verollisen hinnan.
/// Verollinen hinta lasketaan seuraavalla kaavalla:
///
/// VerotonHinta * (1 + Alv)
/// </summary>
/// <returns></returns>
```

TEKIJÄT:

[4]

```
/// <summary>
/// InMemoryAteriaDb.cs
///
/// Luokka käsittelee tietokantayhteyksiä Ateria-luokan
/// olioiden osalta.
///
/// HUOMAA ETTÄ KYSEESSÄ ON MUISTINVARAINEN TIETOKANTATOTEUTUS!
/// TOTEUTUSTAPAAN VOIT KYSYÄ NEUVOA.
///
/// Luokalla on seuraavia ominaisuuksia:
///
/// 1. Hashtable ateriat
/// </summary>
```

```
/// <summary>
/// InMemoryAteriaDb()
///
/// Parametriton rakenta. Alustetaan tietokanta (hashtable)
/// aterioilla, jotta voidaan kokeilla ohjelman toimintoja.
/// </summary>
```

```
/// <summary>
/// List<Ateria> HaeKaikki()
///
/// Metodi palauttaa kaikki ateriat tietokannasta (Hashtable)
/// listana, joka sisältää Ateria-luokan olioita.
/// </summary>
/// <returns></returns>
```

TEKIJÄT:

[5]

```
/// <summary>
/// Poyta.cs
///
/// Luokka käsittelee ohjelmassa esiintyvien pöytien tietoja.
/// Luokan toimintojen tarkempi kuvaus löytyy sen metodien
/// kuvauksista.
///
/// Luokalla on seuraavia ominaisuuksia:
///
/// 1. int Id
/// 2. int PaikkojenMaara
/// 3. Varaustilanne Varaustilanne (enum? miksi hyvä käyttää?)
/// </summary>

/// <summary>
/// Poyta(int tunnus, int paikkojenMaara, Varaustilanne varattu)
///
/// Parametrillinen konstruktori, joka rakentaa Poyta-tyyppisen
/// olion olemassa olevista tiedoista. Parametreina annetaan
/// luokassa esitellyt ominaisuudet.
/// </summary>
/// <param name="tunnus"></param>
/// <param name="paikkojenMaara"></param>
/// <param name="varattu"></param>

/// <summary>
/// void VaraaPoyta()
///
/// Metodi muuttaa olion tilaa siten, että merkitsee
/// pöydän varaustilanteen varatuksi.
/// </summary>

/// <summary>
/// void VapautaPoyta()
///
/// Metodi muuttaa olion tilaa siten, että merkitsee
/// pöydän varaustilanteen vapaaksi.
/// </summary>

/// <summary>
/// bool OnkoVarattu()
///
/// Metodi tarkistaa onko pöytä varattu. Jos pöytä
/// on varattu, palauttaa se arvon true. Muussa
/// tapauksessa palautetaan arvo false.
/// </summary>
/// <returns></returns>
```

TEKIJÄT:

[5]

```
/// <summary>
/// InMemoryPoytaDb.cs
///
/// Luokka käsittelee tietokantayhteyksiä Poyta-luokan
/// olioiden osalta.
///
/// HUOMAA ETTÄ KYSEESSÄ ON MUISTINVARAINEN TIETOKANTATOTEUTUS!
/// TOTEUTUSTAPAAN VOIT KYSYÄ NEUVOA.
///
/// Luokalla on seuraavia ominaisuuksia:
///
/// 1. Hashtable poydat
/// </summary>

/// <summary>
/// InMemoryPoytaDb()
///
/// Parametriton rakenta. Tietokant (Hashtable)a alustetaan
/// pöytien tiedoilla, joilla voidaan kokeilla ohjelman toimintoja.
/// </summary>

/// <summary>
/// Poyta Hae(int tunnus)
///
/// Metodi palauttaa tietokannasta (hashtable) pöydän
/// parametrina annetun pöydän tunnuksen mukaan.
/// </summary>
/// <param name="tunnus"></param>
/// <returns></returns>

/// <summary>
/// List<Poyta> HaeKaikki()
///
/// Metodi palauttaa tietokannasta (hashtable) kaikki
/// pöydät listana, joka sisältää Poyta-olioita.
/// </summary>
/// <returns></returns>

/// <summary>
/// void VaraaPoyta(int tunnus)
///
/// Metodi hakee tietokannasta (hashtable) pöydän parametrina
/// annetulla pöydän tunnuksella ja muuttaa pöydän tilan
/// varatuksi.
/// </summary>
/// <param name="tunnus"></param>
```

TEKIJÄT:

[6]

```
/// <summary>
/// Asiakas.cs
///
/// Luokka käsittelee asiakkaalle tapahtuvia toimenpiteitä.
/// Luokan tarkemmat toiminnot on kuvattu sen metodien kuvauksissa.
/// Asiakkaalla on kaksi ominaisuutta, joista toinen on vakio:
///
/// 1. bool etukuponki
/// 2. const double ETUKERROIN
/// </summary>

/// <summary>
/// LaskeHinta(double tilauksenhintaa)
///
/// Metodi laskee asiakkaalle tilauksen hinnan seuraavin
/// ehdoin:
///
/// 1. Jos asiakkaalla on etukuponki, metodi palauttaa
///    hinnan, joka on laskettu kaavalla
///    "tilauksenhintaa * ETUKERROIN"
/// 2. Jos asiakkaalla ei ole etukuponkia, palautetaan
///    alkuperäinen hinta sellaisenaan.
/// </summary>
/// <param name="tilauksenhintaa"></param>
/// <returns></returns>
```

TEKIJÄT:

[6]

```
/// <summary>
/// BonusAsiakas.cs
///
/// Luokka periytyy Asiakas.cs luokasta ja omii perittävän
/// luokan ominaisuudet ja toiminnot. BonusAsiakas -luokka
/// laajentaa alkuperäistä Asiakas-luokkaa siten, että sitä
/// voidaan käyttää bonusasiakkaan tietojen tallentamiseen.
/// Bonusasiakkaasta tahdotaan tehdä aina merkintä tietokantaan.
///
/// Bonusasiakkaan toiminnot on kuvattu tarkemmin metodien
/// kuvauksissa.
///
/// Bonusasiakkaalla on kaksi ominaisuutta:
///
/// 1. int Id
/// 2. double Etupisteet
/// </summary>

/// <summary>
/// BonusAsiakas()
///
/// Parametrin rakentaja, jossa määritetään
/// alustusarvot bonusasiakkaan ominaisuuksille
/// seuraavasti:
///
/// Id = 0
/// Etupisteet = 0
/// etukuponki = true (Mistä etukuponki ominaisuus tulee?)
/// </summary>

/// <summary>
/// BonusAsiakas(int id, double etupisteet)
///
/// Parametrillinen rakentaja, jota käytetään bonusasiakkaan
/// tietojen käsittelyyn kun se luodaan olemassaolevien
/// tietojen pohjalta. Etukuponki määritetään päälle
/// aina bonusasiakkaan parametrillisessä rakentajassa.
/// </summary>
/// <param name="id"></param>
/// <param name="etupisteet"></param>

/// <summary>
/// LaskeEtupisteet(double hinta)
///
/// Metodi palauttaa etupisteiden määrän sille
/// annetusta hinta-parametrin mukaan. Etupisteiden
/// laskemiseen käytetään seuraavaa yksinkertaista
/// kaavaa:
///
/// Math.Round(hinta / 50, 1);
/// (mitä math round tekee ja mitä tarkoittaa pilkun
/// jälkeen numero 1?)
/// </summary>
/// <param name="hinta"></param>
/// <returns></returns>

/// <summary>
/// KerrytaEtupisteita(double hinta)
///
/// Metodi kasvattaa bonusasiakkaan etupisteitä mutta
/// ei palauta mitään arvoa.
/// </summary>
/// <param name="hinta"></param>
```

TEKIJÄT:

[6]

```
/// <summary>
/// InMemoryBonusAsiakasDb.cs
///
/// Luokka käsittelee tietokantayhteyksiä Bonusasiakas-luokan
/// olioiden osalta.
///
/// HUOMAA ETTÄ KYSEESSÄ ON MUISTINVARAINEN TIETOKANTATOTEUTUS!
/// TOTEUTUSTAPAAN VOIT KYSYÄ NEUVOA.
///
/// Luokalla on seuraavia ominaisuuksia:
///
/// 1. Hashtable asiakkaat
/// 2. static int seuraavaId = 1
/// </summary>
```

```
/// <summary>
/// void TallennaMuutokset(BonusAsiakas asiakas)
///
/// Metodi tallentaa Bonusasiakkaan tietojen muutokset
/// tietokantaan. (hashtable)
///
/// TÄTÄ METODIA EI OLE KÄYTETTY MISSÄÄN
/// </summary>
/// <param name="asiakas"></param>
```

```
/// <summary>
/// void Uusi(BonusAsiakas asiakas)
///
/// Metodi lisää uuden Bonusasiakkaan tiedot tietokantaan. (hashtable)
/// </summary>
/// <param name="asiakas"></param>
```


TEKIJÄT:

[7]

```
/// <summary>
/// Tapahtumantila.cs
///
/// Enum tyyppinen vakio. Ei ole luokka vaan
/// sisältää tiedon niistä vaihtoehtoista, mitä
/// arvoja tapahtuman tila voi saada. Tapahtuman
/// tila voi olla jokin seuraavista:
///
/// 1. Vahvistamaton
/// 2. Vahvistettu
/// </summary>

/// <summary>
/// Varaustilanne.cs
///
/// Enum tyyppinen vakio. Ei ole luokka vaan
/// sisältää tiedon niistä vaihtoehtoista, mitä
/// arvoja Varaustilanne voi saada. Varaustilanteella
/// on kaksi eri tilaa:
///
/// 1. Vapaa
/// 2. Varattu
/// </summary>
```