

# Tuotekuvaus

---

## Sisällysluettelo

Arviointimatriisi .....	2
Yleistä .....	2
Ekstrat.....	3
Ohjelmiston kuvaus ja toteutus.....	3
Luokkakaavio $a_{43}$ .....	4
Luokka Program.....	5
Luokka Omistaja .....	5
Luokka Veroilmoitus .....	7
Luokka Muunimi .....	7
Luokka Asunto1 .....	7
Luokka Asunto2 .....	7
Luokka Lue .....	7
Luokka Suoritus .....	7
Rajapinta IMaksaminen .....	7
Luokka TallennaSumma.....	7
Rajapinta ISumma.....	7
Luokka Pankki .....	8
Luokka Email.....	8
Luokka KokonaisSumma.....	8
Rajapinta InterfaceProsentti .....	8
Luokka Prosentti.....	8
Luokka Osuus.....	8
Luokka Tapahtumarivit.....	9
Luokka Rivit.....	9
Abstrakti luokka Maksutapahtumia .....	9
Rajapinta IAsuntonumero .....	9
Luokka Asuntonumero .....	9
Luokka Viimeinenmaksu.....	9
Rajapinta ITulosta .....	9
Luokka Maksaja .....	10
Luokka Maksu .....	10
Luokka Maksunsaaja.....	10
Dokumentointi.....	10

## Arviointimatriisi

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} =$$

	1	2	3	4	5
Oliot	osaa luoda luokasta olion	osaa luoda dynaamisen oliokokeelman	osaa määritellä oliolle kentät ja metodit	osaa määritellä näkyvyydet julkisiksi ja yksityisiksi	osaa määritellä oliolle property-kenttiä
Luokkarakenne	osaa määritellä luokan, jossa on konstruktori	osaa määritellä luokan, jossa on viite toiseen olioon	osaa määritellä luokan, jossa on viite oliokokeelmaan	osaa määritellä luokan, joka perii yläluokan	osaa määritellä luokan, jossa hyödynnetään metodin ylikirjoittamista
Rajapinnat	osaa määritellä rajapinnan	osaa määritellä rajapinnan toteuttavan luokan	osaa hyödyntää polymorfismia rajapinnan avulla	lähdekoodista on erotettu jokin osa rajapinnan avulla	lähdekoodista on erotettu useampi osa rajapintojen avulla
Dokumentointi	lähdekoodi on luettavaa	lähdekoodi on jäsennelty	lähdekoodista on tehty luokkakaavio	lähdekoodissa on kommentteja	lähdekoodi on itsekomentoiva
Tallentaminen	osaa tallentaa tekstiä tiedostoon	osaa ladata tekstiä tiedostosta	osaa tallentaa olioidatan tiedostoon	osaa ladata olioidatan tiedostosta	osaa hyödyntää olioidatan sarjallistamista

,missä  $m, n \in \{1, 2, 3, 4, 5\}$

Jatkossa viitataan arviointimatriisin alkioihin symboleilla  $a_{11} \dots a_{55}$ , kun esitetään kohtaa, missä alkion osoittama ominaisuus on toteutettu ohjelmistossa.

Arviointimatriisin kaikki alkiot  $a_{11} \dots a_{55}$  on pyritty huomioimaan ja toteuttamaan ohjelmistotuotteessa, mikä tässä harjoitustyössä on tehty tuotekuvauksessa esitetään.

## Yleistä

Työssä esitetään ohjelmisto, mikä toimii (kuvitellusti) vuokra-asuntojen asukkaiden, omistajien ja suoritusten kirjaamisen ja tiedonvälittämisen hallinnan sovelluksena. Tässä ei oteta kantaa siihen eikä oleellista ole kiinnittääkään huomiota siihen, kuinka tarkoituksenmukainen se on todellisen elämän tarpeisiin em. käytössä. Oleellista tässä on ollut ohjelmiston kehityksessä soveltaa olio-ohjelmoinnin kannalta edellä esitetyssä arviointimatriisissa ohjelmointiin liittyviä menetelmiä ja piirteitä.

## Ekstrat

Lisänä edellä arviointimatriisissa mainittujen ohjelmointimenetelmien ja ohjelmoinnin menetelmien keinoin kehitettävien ja toteutettavien toimintojen lisäksi työssä on esimerkin omaisesti esitetty eräänlainen yksinkertainen käyttöliittymä ohjelmiston käynnistämiseksi sekä menetelmä lähettää luokassa esitetyn metodin mukaan tiedostoja sähköpostiosoitteesta toiseen. Näitä ohjelmoinnin menetelmiä ei kurssin luennoilla esitetty.

## Ohjelmiston kuvaus ja toteutus

Ohjelmisto voidaan tyypillisesti käynnistää käyttäen ohjelmoinnin kehitysympäristön esim. Visual Studio 17 Start-ominaisuutta ohjelmiston nimiavaruuden (namespace) ConsoleApp1 luokissa esitettyjen metodien olioiden implementoimiseksi. Ohjelmiston käyttöliittymänä toimii tällöin konsoli ja etenemistä toteutetaan komentorivien kautta.

Toinen mahdollisuus on käynnistää sovellus esim. Windows-käyttöjärjestelmässä kehitysympäristön EXE- (Executable file) ohjelmatiedostomuotoon generoimalla ConsoleApp1.exe-tiedolla, mihin tietokoneohjelma on tallennettu. Käyttöjärjestelmä lataa tiedoston ja suorittaa siinä olevan ohjelmakoodin.

Kolmas keino käynnistää sovellus on tässä työssä **extrana** Visual Studio 17 kehitysympäristössä kehitetty käyttöliittymä siten, että nimiavaruudessa WindowsFormsApp1 olevien luokkien metodit toteuttavat StartForm.cs luokan metodin mukaan edellä mainitun ConsoleApp1.exe-tiedoston suorituksen oliona olevan käyttöliittymänä toimivan StartForm-ikkunan Start-nappulasta klikkaamalla saa tapahtuman toteutumaan.

Sovellus voidaan käynnistää Windows-käyttöjärjestelmässä kehitysympäristön EXE- (Executable file) ohjelmatiedostomuotoon generoimalla WindowsFormsApp1.exe-tiedolla. ConsoleApp1.exe-tiedoston tulee tällöin olla samassa kansiossa kuin WindowsFormsApp1.exe-tiedoston, koska StartForm.cs-luokassa viitataan suoraan ConsoleApp1.exe-tiedostoon eikä niin ollen polkumäärittelyjä samassa kansiossa oleville exe-tiedostoille tarvita.

Oheisessa WindowsFormsApp1.zip-tiedostossa ovat pakattuina kansio WindowsFormsApp1 alikansioineen ja tiedostoineen, mitkä liittyvät em. käyttöliittymään. Lisäksi zip-tiedostossa ovat pakattuina tiedostot WindowsFormsApp1.exe, ConsoleApp1.exe ja tuotespeksi.docx, jotka tarvitaan, kun sovellus otetaan käyttöön Windows-käyttöjärjestelmässä avaamalla käyttöliittymä suoraan WindowsFormsApp1.exe-tiedostosta esim. sitä hiiren vasemmalla näppäimellä tupla-klikkaamalla kursorin osoittaessa tiedostoa. Oheissa myös nimiavaruuden ConsoleApp1-tiedostot zip-tiedostossa ConsoleApp1.zip.

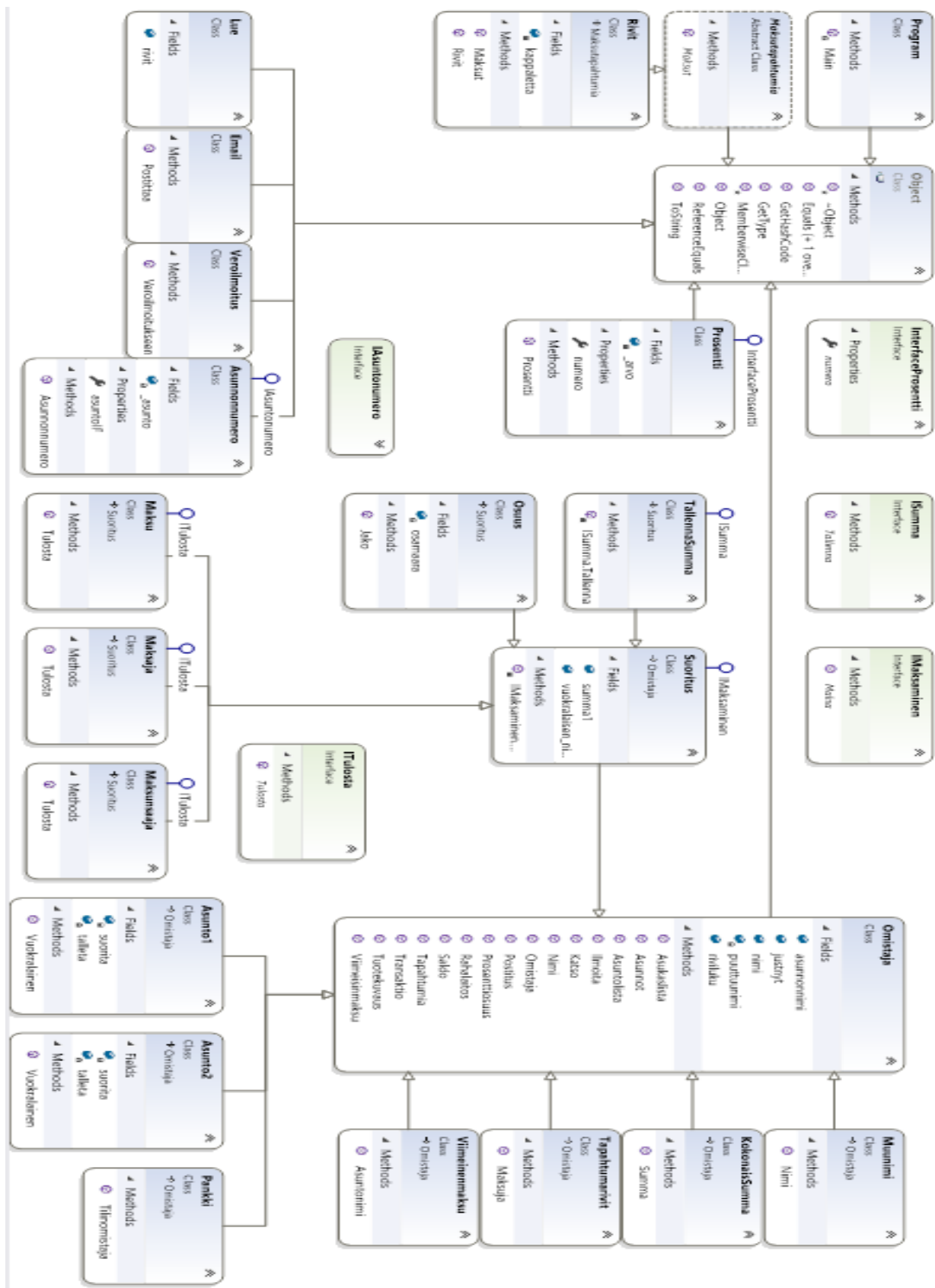


WindowsFormsApp1.  
zip



ConsoleApp1.zip

## Luokkakaavio a43



Työssä toteutetun sovelluksen nimiavaruuden ConsoleApp1luokat ja rajapinnat esitetään seuraavaksi.

### Luokka Program

Tämä on ohjelmiston pääohjelma, missä viittaus NET-kehitysympäristön luokkakirjaston "using System" luokkiin.

Main-metodissa määritellään paikallinen muuttuja omistaja tyyppiä Omistaja ja kutsutaan luokan Omistaja.cs seuraavat metodit:

Tuotekuvaus() toteuttaa tämän dokumentin "tuotespeksi.docx" avaamisen.

Asuntolista(), missä "Dictionary-luokan" mukaan luodaan lista asunnoista.

Asukaslista(), missä on "Dictionary" asukkaiden listaamista varten dynaamisesti olioita lisäten. Samoin rajapinta " IList" ja luokka " List". **a<sub>12</sub>, a<sub>23</sub>,**

Nimi(), missä on menetelmä omistajan nimen kirjoittamiselle.

Asunnot(), missä tehdään asunnonvalinta.

Katso(), missä voidaan valita tilitapahtumien tulostus näytölle.

Rahalaitos(), missä metodi johtaa luokassa Pankki tiedostoon kirjoittamisen metodiin. **a<sub>51</sub>**

Ilmoita() "veroilmoitukseen" ilmoittaminen.

Postitus() johtaa luokassa Email sähköpostilla lähettämisen metodiin Postittaa(). **Extra**

Saldo() johtaa luokassa KokonaisSumma maksujen tiedostossa "summat.txt" yhteen laskemiseen.

Prosenttiosuus() johtaa luokassa KokonaisSumma suoritettavan maksetun osuuden tilin saldosta määrittelyn menetelmään.

Tapahtumia()johtaa luokassa Tapahtumarivit menetelmään tulostaa maksutapahtumien määrä.

Viimeisinmaksu() johtaa luokassa Viimeinenmaksu metodiin Asuntonimi(), jotta tulostetaan asunto, minkä asukkaalta tullut viimeisin maksusuoritus.

Transaktio() johtaa tiedoston Maksutiedot.cs luokkiin Maksaja, Maksu ja Maksunsaaja, missä on hyödynnetty polymorfismia rajapinnan avulla. **a<sub>33</sub>**

### Luokka Omistaja

Tässä luokassa ovat viittaukset NET-kehitysympäristön luokkakirjastojen `using System;`  
`using System.IO;`  
`using System.Runtime.Serialization.Formatters.Binary;`  
`using System.Collections.Generic;`  
luokkiin.

Luokassa on mm. konstruktori `public Omistaja()`, missä parametrille "puuttuunimi" annetaan arvoksi merkkijono "Nimi antamatta", mikäli pyydettyä omistajan nimeä ei anneta. Sen jälkeen poistutaan sovelluksesta.

Myös seuraavat kentät on luokassa määritelty:

```
public static string nimi; //Näkyvyys julkinen.  
public static DateTime justnyt; //Näkyvyys julkinen.  
public static string asunnonnimi; //Näkyvyys julkinen.  
public static int riviluku; //Näkyvyys julkinen.  
private string puuttuunimi; //Näkyvyys ei ole julkinen eli on yksityinen. a13, a14
```

Muutujan tyyppiksi on määritelty "static", kun halutaan Omistaja-luokan metodissa sille määritellyn arvon periytyvän aliluokalle.

### Luokan metodeissa **a<sub>13</sub>**

- **Tuotekuvaus()** avataan tämä Tuotekuvaus-tiedosto "tuotespeksi.docx"
- **Asuntolista()** listataan näytölle metodissa määritellyt tektirivit
- **Asukaslista()** kirjoitetaan tiedostoon "asukkaat.txt" ja listataan näytölle mm. asukkaiden nimiä
- **Nimi()** kirjoitetaan staattiselle muuttujalle "nimi" merkkijonona arvoksi omistajan nimi, metodi on määritelty virtual-tyyppiseksi, jotta aliluokassa Muunimi se voidaan ylikirjoittaa **a<sub>25</sub>**
- **Ilmoita()** johtaa luokkaan Veroilmoitus, missä on menetelmä olemassa olevan "pankkitili.txt" rivien kopioimiseksi tiedostoon "veroilmoitus.txt",  
" Veroilmoitus laadi = new Veroilmoitus(); //Tässä luodaan uusi olio luokkaan Veroilmoitus."
- **Asunnot()** -metodissa valitaan joko luokan Asunto1 tai luokan Asunto2 metodi Vuokralainen, mistä on linkitykset luokkaan Suoritus kirjoitettavan maksun tallentamiseksi tiedostoon "pankkitili.txt" ja luokkaan TallennaSumma em. maksun tallentamiseksi tiedostoon "summa.txt" **a<sub>11</sub>**
- **Katso()** on menetelmä tilitietojen tulostamiseksi konsolin näytölle siten, että siinä hyödynnetään oliodatan sarjallistamista, sarjallistavassa luokassa Lue rivit tiedostosta "pankkitili.txt" luetaan ja binäärimuodossa kirjoitetaan tiedostoon "tiedosto.txt", mistä ne taas luetaan ja puretaan sarjallistuneisuus (Deserialize) minkä jälkeen ne jälleen tekstimuodossa riveittäin tulostetaan näytölle ja vielä lasketaan tulostettujen rivien määrä **a<sub>53</sub>, a<sub>54</sub>, a<sub>55</sub>**
- **Rahalaitos()**, missä linkitys luokan Pankki metodiin tilinomistajien nimien sekä tapahtuma-ajankohtien kirjaamiseksi tiedostoon "tilinomistajat.txt"
- **Postitus()**, missä linkitys luokkaan Email, missä metodi Postittaa() sähköpostin lähettämiseksi määritellystä sähköpostiosoitteesta määritellyyn sähköpostiosoitteeseen

### Extra

- **Saldo()**, missä linkitys luokkaan KokonaisSumma ja sen metodiin Summa() missä menetelmä toteuttaa kirjattujen maksujen tiedostossa "summat.txt" yhteen laskemisen
- **Prosenttiosuus()**, missä linkitys metodiin Jako() luokassa Osuus tiedostossa Prosentti.cs, missä menetelmä toteuttaa maksetun osuuden määrittelyn prosenttilukuna tilin saldosta ja tulostaa viimeisimmän maksetun summan, tiedoston Prosentti.cs luokassa Prosentti määritelty ominaisuus "numero" **a<sub>15</sub>**
- **Tapahtumia()**, missä linkitys metodiin Maksuja() luokassa Tapahtumarivit, missä toteutetaan maksutapahtumien määrän tulostaminen näytölle
- **Viimeisinmaksu()**, missä linkitys metodiin Asuntonimi() luokassa Viimeinenmaksu, mikä toteuttaa sen asunnon tulostuksen näytölle liittyen viimeisimpään maksusuorituksen, luokka Viimeinenmaksu on tiedostossa Asunnonnumero.cs **a<sub>31</sub>, a<sub>32</sub>, a<sub>34</sub>, a<sub>35</sub>**
- **Transaktio()**, missä linkitys rajapinnassa ITulosta esitettyyn metodiin Tulosta() tiedostossa Maksutiedot.cs esitettyihin luokkiin Maksaja, Maksu ja Maksunsaaja, joissa on hyödynnetty polymorfismia rajapinnan avulla ja toteutetaan oliot maksajasta "vuokralaisen\_nimi", maksusta "summa1" ja maksunsaajasta "nimi"

Seuraavien luokissa käytettyjä luokkakirjastoja ei mainita tässä Tuotekuvaus-dokumentissa, mutta ne ovat nähtävissä ko. luokkien ohjelmakoodeissa.

### Luokka Veroilmoitus

Tässä luokassa metodi Veroilmoitukseen() toteuttaa tiedoston "pankkitili.txt" kopioinnin tiedostoon "veroilmoitus.txt".

### Luokka Muunimi

Luokka Muunimi perii luokan Omistaja. **a<sub>24</sub>**. Luokassa Muunimi toteutetaan Luokan Omistaja metodissa "public virtual void Nimi()" toteutetun nimen kirjoituksen ylikirjoittaminen metodilla "public override void Nimi()".

### Luokka Asunto1

Luokka Asunto1 perii luokan Omistaja. **a<sub>24</sub>**

Luokassa Asunto1 on metodi Vuokralainen(), mikä toteuttaa luokan Suoritus metodin IMaksaminen.Maksa() sekä luokan TallennaSumma metodin ISumma.Tallenna(). Luokka Suoritus perii rajapinnan IMaksaminen ja luokka TallennaSumma rajapinnan ISumma. **a<sub>32</sub>**

### Luokka Asunto2

Luokka Asunto2 perii luokan Omistaja. **a<sub>24</sub>**

Myös luokassa Asunto2 on metodi Vuokralainen(), mikä toteuttaa luokan Suoritus metodin IMaksaminen.Maksa() sekä sekä luokan TallennaSumma metodin ISumma.Tallenna(). **a<sub>32</sub>**

### Luokka Lue

Luokka Lue on serialisoituva [Serializable] ja siinä toteutetaan tiedoston "pankkitili.txt" rivien lukeminen. **a<sub>55</sub>**

### Luokka Suoritus

Luokka Suoritus perii luokan Omistaja. **a<sub>24</sub>** ja toteuttaa rajapinnassa IMaksaminen esitetyn metodin Maksa(), missä edelleen metodi IMaksaminen.Maksa() toteuttaa tiedostoon "pankkitili.txt" olemassa olevien rivien perään uuden rivin kirjoittamisen.

### Rajapinta IMaksaminen

Katso edellinen kohta "Luokka Suoritus"!

### Luokka TallennaSumma

Luokka Suoritus perii luokan Suoritus. **a<sub>24</sub>** ja toteuttaa rajapinnassa ISumma esitetyn metodin Tallenna(), missä edelleen metodi ISumma.Tallenna() toteuttaa tiedostoon "summat.txt" olemassa olevien rivien perään uuden rivin kirjoittamisen.

### Rajapinta ISumma

Katso edellinen kohta "Luokka TallennaSumma"!

## Luokka Pankki

Luokka Pankki perii luokan Omistaja. **a<sub>24</sub>**

Metodissa Tilinomistaja() toteutetaan annetun omistajan nimen ja tapahtuma-ajankohdan tallennus tiedostoon "tilinomistajat.txt", siten, että viimeisin tallennus kirjautuu tiedostoon uutena rivinä edellisen tallennuksen jälkeen. **a<sub>51</sub>**

## Luokka Email

Luokka Email on tässä työssä ja sovelluksessa ylimääräisenä harjoituksena. Sitä ei ole huomioitu arviointimatriisissa. Tämä on kuitenkin esimerkkinä verkko-ohjelmoinnin mahdollisuuksista, mikä on tärkeä osa ohjelmointi- ja tietotekniikkaa. Luokassa Email metodissa Postittaa() toteutetaan tiedoston "tilinomistajat.txt" sähköpostitus lähettäjän sähköpostiosoitteesta vastaanottajan sähköpostiosoitteeseen. **Extra**

## Luokka KokonaisSumma

Luokka KokonaisSumma perii luokan Omistaja. **a<sub>24</sub>**

Metodissa Summa() toteutetaan tiedoston "summat.txt" rivien lukeminen ja sen jälkeen luettujen rivien määrä lasketaan. **a<sub>52</sub>**

## Rajapinta InterfaceProsentti

Rajapinta InterfaceProsentti määrittelee ominaisuuden "numero" tyyppiä "double". Tämä toimii rajapintana, joka toteutetaan luokassa Prosentti ja on luokassa Osuus metodin Jako() parametrin "maksu" tyyppimäärittely.

## Luokka Prosentti

Luokan Prosentti rajapinta on InterfaceProsentti ja tämä luokka toteuttaa rajapinnan.

Luokan konstruktorissa annetaan parametrille "\_arvo" arvoksi ominaisuus "numero". **a<sub>21</sub>**

## Luokka Osuus

Luokka Osuus on toisena luokkana tiedostossa Prosentti.cs ja se perii luokan Suoritus ja sieltä parametrin "summa1" arvon.

Luokassa metodi Jako() toteuttaa viimeisimmän maksun osuuden laskennan prosentteina kaikista suoritetuista maksuista. **a<sub>22</sub>, a<sub>24</sub>**



## Luokka Tapahtumarivit

Luokka Tapahtumarivit tiedostossa Tapahtumarivit.cs perii luokan Omistaja. **a<sub>24</sub>**  
Luokassa Omistaja metodi Katso() tuottaa tiedoston "tiedosto.txt" rivien määrän laskennan, mikä annetaan arvoksi parametrille "riviluku". Abstraktissa luokassa Maksutapahtumia on esitetty metodi Maksut(), mikä toteutetaan ja ylikirjoitetaan **luokassa Rivit** tiedostossa Tapahtumarivit.cs ja palautetaan luokan Tapahtumarivit metodille Maksuja(), jotta voidaan toteuttaa parametrin "riviluku" ilmoittama maksutapahtumien määrän tulostus konsolin näytölle. **a<sub>22</sub>**

## Luokka Rivit

Luokka Rivit perii abstraktin luokan Maksutapahtumia. Luokan Rivi tehtävä määritelty edellä kohdassa "Luokka Tapahtumarivit". **a<sub>21</sub>**

## Abstrakti luokka Maksutapahtumia

Abstrakki luokka Maksutapahtumia esitelty edellä kohdassa "Luokka Tapahtumarivit".

## Rajapinta IAsuntonumero

Rajapinta IAsuntonumero määrittelee ominaisuuden "asuntoIF" tyypiksi "string". Tämä toimii rajapintana, joka toteutetaan luokassa Asuntonumero ja on luokassa Viimeinenmaksu metodin Asuntonimi() parametrin "asuntonimi" tyyppimäärittely.

## Luokka Asuntonumero

Luokan Asuntonumero rajapinta on IAsuntonumero ja tämä luokka toteuttaa rajapinnan. Luokan konstruktorissa annetaan parametrille "\_asunto" arvoksi ominaisuus "asuntoIF". **a<sub>21</sub>**

## Luokka Viimeinenmaksu

Luokka Viimeinenmaksu on toisena luokkana tiedostossa Asuntonumero.cs ja se perii luokan Omistaja ja sieltä parametrin "asunnonnimi" arvon. Luokassa metodi Asuntonimi() toteuttaa tulostuksen, missä kerrotaan asunto, mihin viimeisin maksusuoritus liittyy. **a<sub>22</sub>, a<sub>24</sub>**

## Rajapinta ITulosta

Rajapinnassa esitetään metodi Tulosta(). **a<sub>33</sub>**

### Luokka Maksaja

Luokka Maksaja tiedostossa Maksutiedot.cs perii luokan Suoritus ja toteuttaa rajapinnan ITulosta. Metodi Tulosta() toteuttaa maksajan (asukkaan) nimen tulostamisen näytölle. **a<sub>33</sub>**

### Luokka Maksu

Luokka Maksu tiedostossa Maksutiedot.cs perii luokan Suoritus ja toteuttaa rajapinnan ITulosta. Metodi Tulosta() toteuttaa maksetun summan tulostamisen näytölle. **a<sub>33</sub>**

### Luokka Maksunsaaja

Luokka Maksunsaaja tiedostossa Maksutiedot.cs perii luokan Suoritus ja toteuttaa rajapinnan ITulosta. Metodi Tulosta() toteuttaa maksun saajan (omistajan) nimen tulostamisen näytölle. **a<sub>33</sub>**

### Dokumentointi

Koodin osalta arviointimatriisin alkioden **a<sub>41</sub>**, **a<sub>42</sub>**, **a<sub>43</sub>**, **a<sub>44</sub>** ja **a<sub>45</sub>** määrittelyt on huomioitu ja toteutettu.