

1. UML –mallinnuskielen luokkakaavio

- Yleistä UML-mallinnuskielestä https://en.wikipedia.org/wiki/Unified_Modeling_Language
- Luokkakaavio on staattinen mallinnustyyppi.
- Luokkakaavio kuvaa järjestelmän staattisen eli pysyvän rakenteen luokkien ja niiden välisten suhteiden avulla.
- Luokkakaavion yksi tehtävä on määrittää perusta muille kaavioille, kuten olioiden tiloja (tilakaavio) ja olioiden välistä yhteistyötä kuvaaville (viestiyhteyksikaavio) dynaamisille kaavioille, jotka kuvaavat järjestelmän muita piirteitä.
- Luokkakaavion luokka voidaan toteuttaa suoraan oliosuuntautuneella ohjelmointikielellä (C++, Java, C#, Python ...), joka tukee suoraan luokka-rakennetta.
- Luokkakaavion tekemistä varten luokat on tunnistettava ja kuvattava. Kun useampia luokkia on olemassa, niiden välille voidaan muodostaa erilaisia suhteita/yhteyksiä.

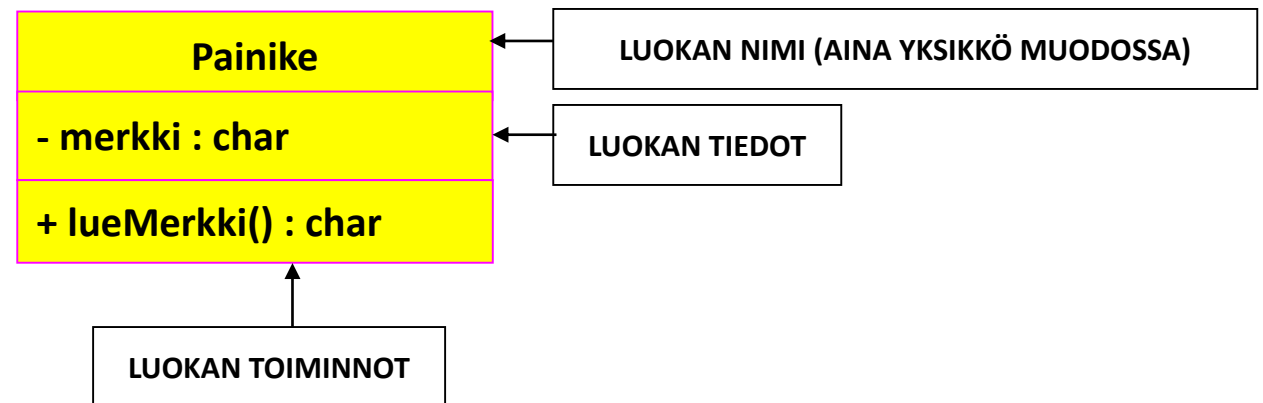
1. UML –mallinnuskielen luokkakaavio

- Luokkakaavion luokan kuvaus sisältää seuraavat osat:
 - luokan nimi, jäsenmuuttujat eli tiedot (=data, attribuutit) ja jäsenfunktiot eli toiminnot (=funktiot/metodit)
- Luokkalaatikon ylimmäinen osa sisältää luokan nimen. Nimi tulisi johtaa ongelma-alueelta ja sen olisi oltava mahdollisimman yksikäsitteinen.
- Luokilla on attribuutteja eli tietokenttiä, jotka kuvaavat olioiden ominaisuuksia ja tiloja.
- Luokilla on jäsenfunktioita (=operaatioita/toimintoja/metodeja), jotka kuvaavat olioiden toiminnallisuutta.
- Attribuuteilla ja toiminnoilla on erilaisia näkyvyyksiä: **public**, **protected** ja **private**.
- Case –työkalussa (esim. draw.io) näkyvyys esitetään usein seuraavin merkein: public +, private – ja protected #
- Seuraavilla sivuilla esimerkkejä luokan kuvaamisesta UML-notaation mukaisesti.

1. UML –mallinnuskielen luokkakaavio

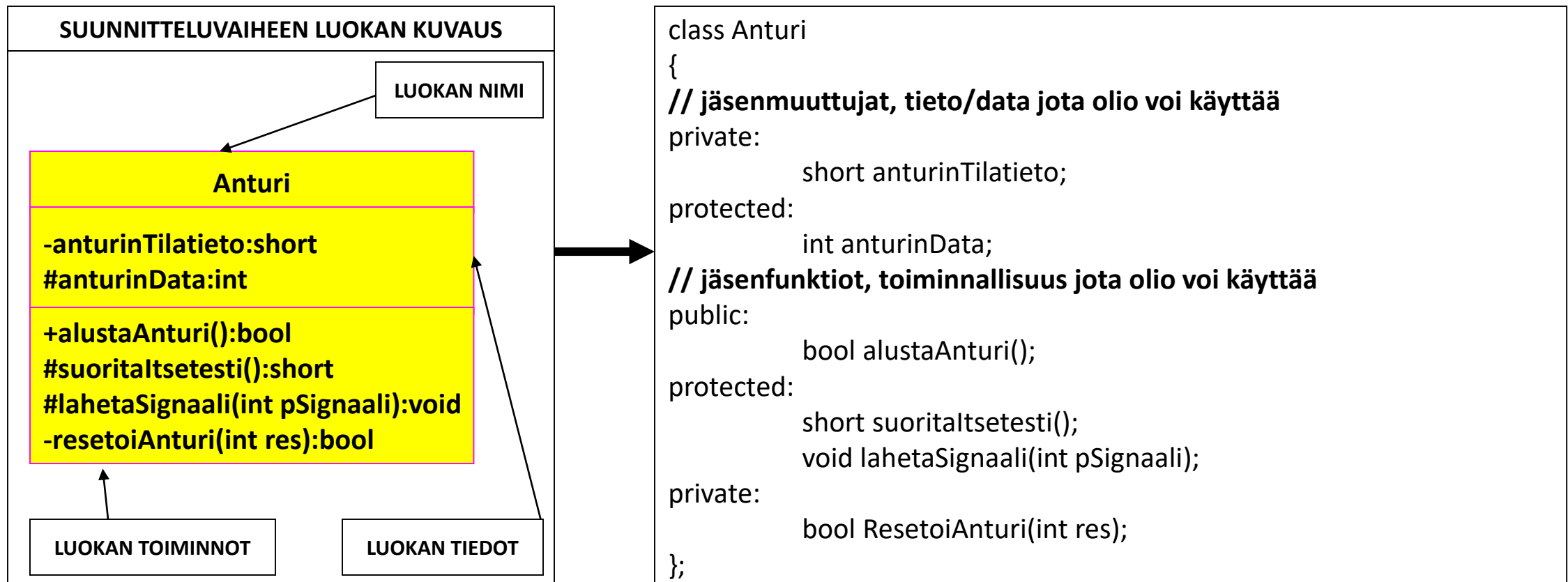
Luokan kuvaaminen UML notaation mukaan

- Luokka piirretään suorakaiteena, joka on jaettu kolmeen osaan; luokan nimi, luokan tiedot ja luokan toiminnallisuus.
- Nimiosassa on luokan nimi **yksikössä**. Luokan nimi tulisi johtaa ongelma-alueelta ja sen olisi oltava mahdollisimman yksikäsitteinen.
- Luokan tiedot esitetään jäsenmuuttujina ja luokan toiminnot esitetään jäsenfunktioina.



Luokan kuvauksesta ohjelmakoodiksi

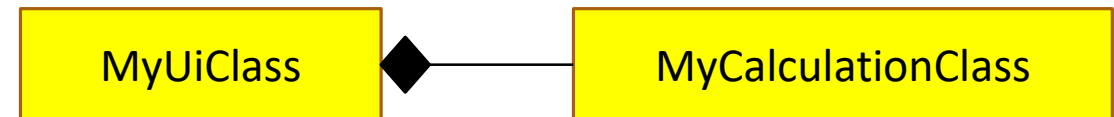
- Luokan kuvauksesta ohjelmakoodiksi
 - julkinen (public +)
 - yksityinen (private -)
 - suojattu (protected #)



Olio-ohjelmointi: olioiden yhteistyö, vahva kooste

Tehtävä 1: Vahva kooste, MyCompositionCalculator

- Luo Qt Creatorilla uusi projekti **MyCompositionCalculator** (**Non-Qt Project->Plain C++ Application**) ja lisää projektiin luokat **MyUiClass** (=koosteluokka) ja **MyCalculationClass** (=osaluokka)
- Jos luokkien rakenteeseen (.h tiedostot) ei tule automaattisesti muodostin- ja tuhoajafunktiota, niin lisää ne sinne.
- Lisää luokkien toteutukseen (.cpp tiedostot) muodostin- ja tuhoajafunktioiden toteutuksien rungot.



Olio-ohjelmointi: olioiden yhteistyö, vahva kooste

Tehtävä 1: Vahva kooste, MyCompositionCalculator

- Ohjelmassa on tarkoitus, että koosteluokkaan **MyUiClass** rakennetaan nelilaskimen käyttöliittymä (UI=**U**ser **I**nterface).
- Osaluokkaan **MyCalculationClass** rakennetaan jäsenfunktiot, joilla hoidetaan laskutoimituksien suorittaminen ja palautetaan laskutoimituksen tulos näytettäväksi koosteluokalle.
- Eli rakennetaan luokista luotavat oliot **yhdessä** tekemään ohjelman haluttu toiminnallisuus.
- Suunniteltaessa luokkarakenteita kannattaa aina ottaa lähtökohdaksi, että luokalla on vain yksi tehtäväkokonaisuus ohjelmassa.
- *Tässä esimerkissä toisella luokalla on tehtävänä näyttää näytöllä ohjelmassa asioita, ja toisella luokalla suorittaa ohjelman vaatimat laskutoimitukset.*
- Samaa suunnitteluperiaatetta kannattaa noudattaa isommissakin ohjelmissa.
- Luokalla on vastuu esimerkiksi: tietokantaoperaatioista, tietoturvasta, laitteen tietojen lukemisesta, käyttöliittymästä jne.

Olio-ohjelmointi: olioiden yhteistyö, vahva kooste

Tehtävä 1: Vahva kooste, MyCompositionCalculator

- Nelilaskimen valikon käyttöliittymä on kuten kuvassa oikealla.
- Kun käyttäjä on valinnut laskutoimituksen, niin näytöllä kysytään käyttäjältä kaksi desimaalilukua, jonka jälkeen laskutoimitus suoritetaan ja vastaus näytetään näytöllä kuten kuvassa oikealla.
- Kun vastaus on näytetty, niin valikko tulostetaan uudelleen näytölle.

```
NELILASKIN
=====
1 = Summa
2 = Erotus
3 = Tulo
4 = Osamaara
5 = Lopeta
Valitse: 1

Anna luku yksi:1.5
Anna luku kaksi:2.3

Laskutoimitus ja tulos:
1.5+2.3=3.8

NELILASKIN
=====
1 = Summa
2 = Erotus
3 = Tulo
4 = Osamaara
5 = Lopeta
Valitse: 5
Press <RETURN> to close this window...
```

IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

Olio-ohjelmointi: olioiden yhteistyö, vahva kooste

Tehtävä 1: Vahva kooste, MyCompositionCalculator

- Lisää koosteluokan **private** osaan koosteininen yhteys osaluokkaan seuraavasti

MyCalculationClass objectMyCalculationClass;

- Lisää koosteluokkaan jäsenmuuttujat **double numberOne, numberTwo, result;**
- Lisää koosteluokkaan jäsenfunktio **void inputNumbers();** jossa pyydetään käyttäjää syöttämään luvut. Koodaa jäsenfunktion toteutus alla olevan mukaisesti

```
void MyUiClass::inputNumbers()
{
    cout << "Anna luku yksi:";
    cin >> numberOne; // cin oliolla voidaan pyytää käyttäjää syöttämään muuttujalle arvo
    cout << "Anna luku kaksi:";
    cin >> numberTwo;
}
```


IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

Olio-ohjelmointi: olioiden yhteistyö, vahva kooste

Tehtävä 1: Vahva kooste, MyCompositionCalculator

- Lisää koosteluokkaan jäsenfunktio **void showMenu()**; jossa tehdään seuraavat asiat:
 - Näytetään ohjelman valikko
 - Kysytään käyttäjältä valinta. Tämä voidaan tehdä **cin** oliolla/käskyllä seuraavasti:

cin >> menuVariable; // menuVariable on short tyyppinen paikallinen muuttuja

- Hypätään jäsenfunktioon **void inputNumbers()**; jos käyttäjä valitsee arvon 1-4 valikon mukaisesti. Koodaa tämän jälkeen **switch – case** rakenne, jonka mukaan kutsutaan osaolion **objectMyCalculationClass** avulla osaluokan jäsenfunktiota sen mukaan, minkä toiminnon käyttäjä valikosta valitsi.
 - Osaluokan jäsenfunktion laskutoimituksen palauttama arvo asetetaan arvoksi jäsenmuuttujalle **result**.
- Lisää koosteluokkaan jäsenfunktio **void showResult(short parameter)**; jossa vastaus näytetään aikaisemmin esitellyn käyttöliittymän mukaisesti.

Olio-ohjelmointi: olioiden yhteistyö, vahva kooste

Tehtävä 1: Vahva kooste, MyCompositionCalculator

- Lisää osaluokkaan jäsenfunktio **double sum(double parameterOne, double parameterTwo);** ja koodaa jäsenfunktio siten, että se palauttaa sille välitettyjen parametrien summan arvon.
- Lisää osaluokkaan muutkin laskutoimituksia suorittavat jäsenfunktiot.
- **main()** -funktiossa luodaan koosteolio ja kutsutaan jäsenfunktiota **showMenu();**
- **Rakenna ohjelma valmiiksi ja testaa sitä.**

IN00BQ93 Laite- ja tuotesuunnittelun syventävät opinnot: ohjelmoinnin jatkokurssi

Olio-ohjelmointi: olioiden yhteistyö, vahva kooste

Tehtävä 2: Vahva kooste

- Muuta tehtävän 1 ratkaisua siten, että ohjelmassa käytetään dynaamistaa muistinhallintaa.