## 4. Luokkakokonaisuuden muodostaminen

1. Luokkien hahmottaminen
2. Esimerkki: `SuunnattuGraafi`

## Luokkakokonaisuuden muodostaminen

- Tehtävän jakaminen osiin
  - asiakasrelaatio
  - periytymisrelaatio


## Luokkien hahmottaminen

Ongelma → Analyysi → Malli → Luokat → Rutiinit

- Jokainen luokka rakentuu aina sen sisältämän tiedon ympärille
- Keskeistä tietosisältöä käsittelevät rutiinit sijoitetaan samaan luokkaan
  - piirre primitiivisimpään relevanttiin asiakasluokkaan

## Luokkakategoria

- Analyysiluokka
  - suoraan tehtävän määrittelystä
  - kuvaa abstraktia tai konkreettia käsitettä
- Suunnitteluluokka
  - ei suoraa vastinetta ongelmamaailmaan
  - helpottaa toteutusta (esim. iteraattorit)
- Toteutusluokka
  - toistuvasti tarvittava "alimman" tason luokka
  - esim. tietorakenteet

## Esimerkki: Puhelinluettelo

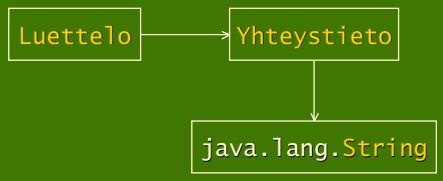
Puhelinluetteloon tallennetaan tiedot
- henkilön nimestä
- yhdestä puhelinnumerosta

Luetteloon voidaan
- lisätä tietoja
- poistaa tietoja
- hakea tietoja antamalla henkilön nimi

## Toimintasuunnitelma

i. Hahmotellaan tarvittavat luokat ja luokkarakenteet.
ii. Määrittellään luokissa tarvittavat luonti-, havainto- ja muutosoperaatiot. Mietitään määrittelyt sekä asiakkaan että toteuttajan kannalta.
iii. Kirjoitetaan operaatioiden määrittelyt (unohtamatta alku- ja loppuehdoja!).
iv. Laaditaan määritellyille luokille toteutukset ja testataan ne.

## i. Luokkakokonaisuuden määrittely




## ii. Luokkien määritttely

- Luokat
  - Yhteystieto
  - Luettelo
- Operaatiot
  - konstruktorit
  - havainnointioperaatiot
  - muunnosoperaatiot

## Yhteystieto-luokka

- konstruktori
  - `public Yhteystieto(String nimi, String numero)`
- havainnointioperaatiot
  - `public String annaNimi()`
  - `public String annaNumero()`
- muunnosoperaatiot
  - `public void asetaNimi(String nimi)`
  - `public void asetaNumero(String numero)`

## Luettelo-luokka: konstruktori

- `public Luettelo()`

## Luettelo-luokka: havainnointioperaatiot

- `public Yhteystieto[] annaYhteystiedot(String nimi)`
- `public int annaYhteystietojenLukumäärä()`
- `public int annaYhteystietojenLukumäärä(String nimi)`
- `public boolean sisältääHenkilön(String nimi)`
- `public boolean sisältääYhteystiedon(String nimi, String numero)`

## Luettelo-luokka: muunnosoperaatiot

- `public void lisääYhteystieto(String nimi, String numero)`
- `public void poistaYhteystieto(String nimi, String numero)`
- `public void muutaYhteystieto(String nimi, String vanhaNumero, String uusiNumero)`
- `public void poistaHenkilö(String nimi)`

## iii. Operaatioiden määrittely

- Määritellään alku- ja loppuehdot
- Määritellään julkinen luokkainvariantti
- Tuloksena julkinen liitäntä

## Yhteystieto 1(2)

```
//-- Konstruktorit
 /** @.pre  nimi != null & numero != null
  *  @.post annaNimi() == nimi &
  *         annaNumero() == numero */
 public Yhteystieto(String nimi, String numero)

 //-- Havainnointioperaatiot
 /** @.pre  true
  *  @.post RESULT == (nimi) */
 public String annaNimi()

 /** @.pre  true
  *  @.post RESULT == (numero) */
 public String annaNumero()
```

## Yhteystieto 2(2)

```
//-- Muunnnosoperaatiot
 /** @.pre  nimi != null
  *  @.post annaNimi() == nimi */
 public void asetaNimi(String nimi)

  /** @.pre  numero != null
   *  @.post annaNumero() == nimi */
 public void asetaNumero(String numero)

/**
 *  @.classInvariant annaNimi() != null &
 *                   annaNumero() != null
 */
```

## Luettelo 1(4)

```
//-- Konstruktorit
 /** @.pre  true
  *  @.post annaYhteystietojenLukumäärä() == 0 */
 public Luettelo()

 //-- Havainnointioperaatiot
 /** @.pre  nimi != null && sisältääHenkilön(nimi)
  *  @.post RESULT.length ==
  *           annaYhteystietojenLukumäärä(nimi) &
  *  FORALL(t : RESULT; t.annaNimi().equals(nimi)) */
 public Yhteystieto[] annaYhteystiedot(String nimi)

 /** @.pre  true
  *  @.post RESULT == (yhteystietojen lukumäärä) */
 public int annaYhteystietojenLukumäärä()
```

## Luettelo 2(4)

```
/**  @.pre  nimi != null
  *  @.post RESULT ==
  *      (annetun henkilön yhteystietojen määrä) */
 public int annaYhteystietojenLukumäärä(
                                    String nimi)

/** @.pre  nimi != null & numero != null
  *  @.post RESULT ==
  *    (annaYhteystietojenLukumäärä(nimi) >= 1) */
 public boolean sisältääHenkilön(String nimi)

/** @.pre  nimi != null & numero != null
  *  @.post RESULT ==
  *          (annetu yhteystieto on luettelossa) */
 public boolean sisältääYhteystiedon(String nimi,
                                     String numero)
```

## Luettelo 3(4)

```
//-- Muunnosoperaatiot
 /** @.pre  (nimi != null & numero != null) &&
  *         !sisältääYhteystiedon(nimi, numero)
  *  @.post this.poistaYhteystieto(nimi,
  *                 numero).equals(OLD(this)) */
 public void lisääYhteystieto(String nimi,
                              String numero)

 /** @.pre  (nimi != null & numero != null) &&
  *         sisältääYhteystiedon(nimi, numero)
  *  @.post this.lisääYhteystieto(nimi,
  *                 numero).equals(OLD(this)) */
 public void poistaYhteystieto(String nimi,
                               String numero)
```

## Luettelo 4(4)

```
/** @.pre  (nimi != null & vanhaNumero != null &
  *        uusiNumero != null) &&
  *        sisältääYhteystiedon(nimi, vanhaNumero)
  *  @.post this.poistaYhteystieto(nimi,
  *         vanhaNumero).lisääYhteystieto(nimi,
  *         uusiNumero).equals(OLD(this))   */
 public void muutaYhteystieto(String nimi,
             String vanhaNumero, String uusiNumero)

 /** @.pre  nimi != null && sisältääHenkilön(nimi)
  *  @.post !sisältääHenkilön(nimi) */
 public void poistaHenkilö(String nimi)

/** @.classInvariant
 *        annaYhteystietojenLukumäärä() >= 0 */
```

## iv. Toteutus

- Yhteystieto
  - kaksi String-tyyppistä jäsenmuuttujaa
- Luettelo
  - Yhteystieto[] taulukko
  - List<Yhteystieto> lista
    - ArrayList<Yhteystieto>
    - LinkedList<Yhteystieto>
  - apurakenteita hakujen nopeuttamissen?

## Privaatit luokkainvariantit ja abstraktiofunktiot

```
/**
 * @.privateClassInvariant (nimi on tallennettu
 *       jäsenmuuttujaan nimi ja numero on
 *       tallennettu jäsenmuuttujaan numero)
 * @.abstractionFunction nimi = (yhteystiedon nimi)
 *       & numero == (yhteystiedon numero)
 */

/**
 * @.privateClassInvariant (yhteystiedot on
 *       tallennettu jäsenmuuttujaan taulukko) &
 *       (annaYhteystietojenLukumäärä() ==
 *        taulukko.length)
 * @.abstractionFunction taulukko == (yhteystiedot)
 */
```