

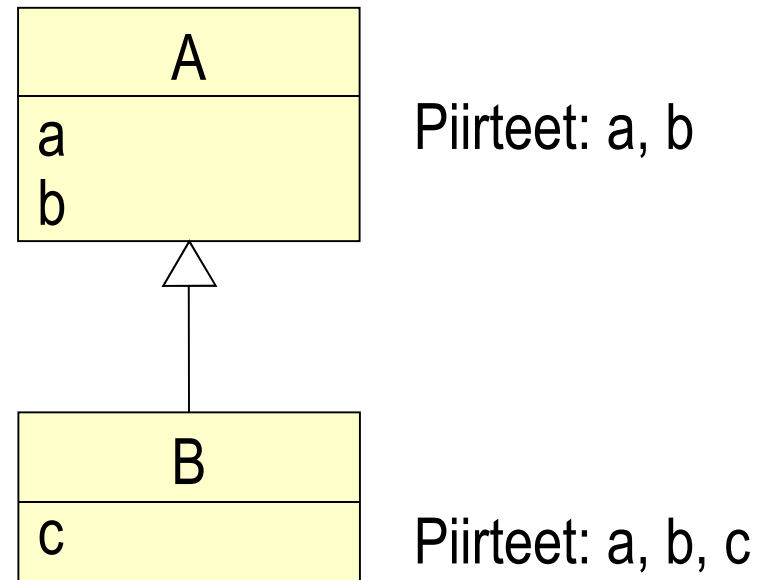
8. Periytyminen

Sisällys

- Mitä on periytyminen?
- Yksittäis- ja moniperiytyminen.
- Oliot ja perityt luokat.
- Filosofinen ja käytännönläheinen näkökulma periytymiseen.
- Periytymisen soveltaminen.

Mitä on periytyminen?

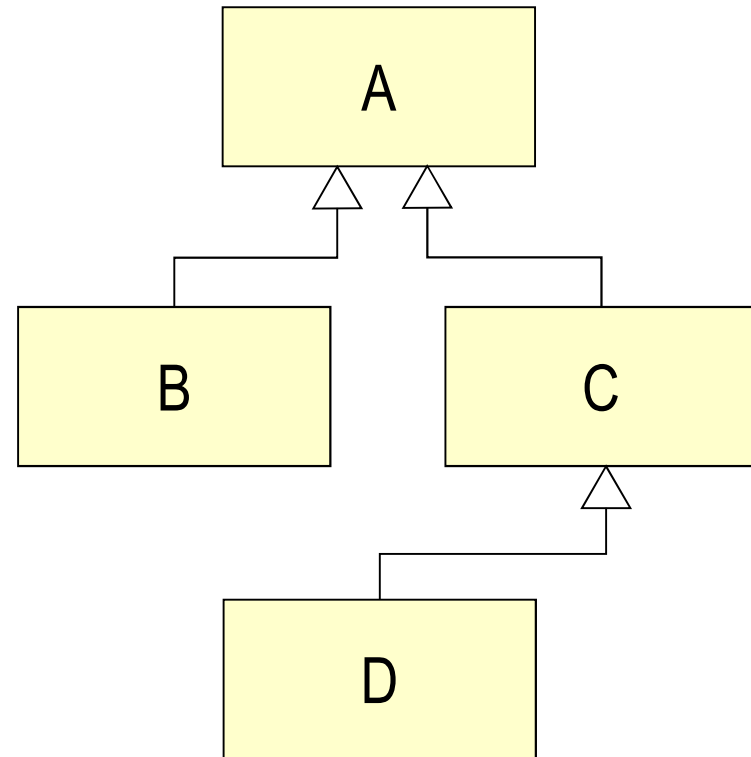
- Periytyminen (inheritance) tarkoittaa sitä, että luokka *B* saa käyttöönsä osan tai kaikki luokan *A* piirteistä.
- Tällöin sanotaan, että *A* on *B*:n **yliluokka** (superclass) ja vastaavasti *B* on *A*:n **aliluokka** (subclass).



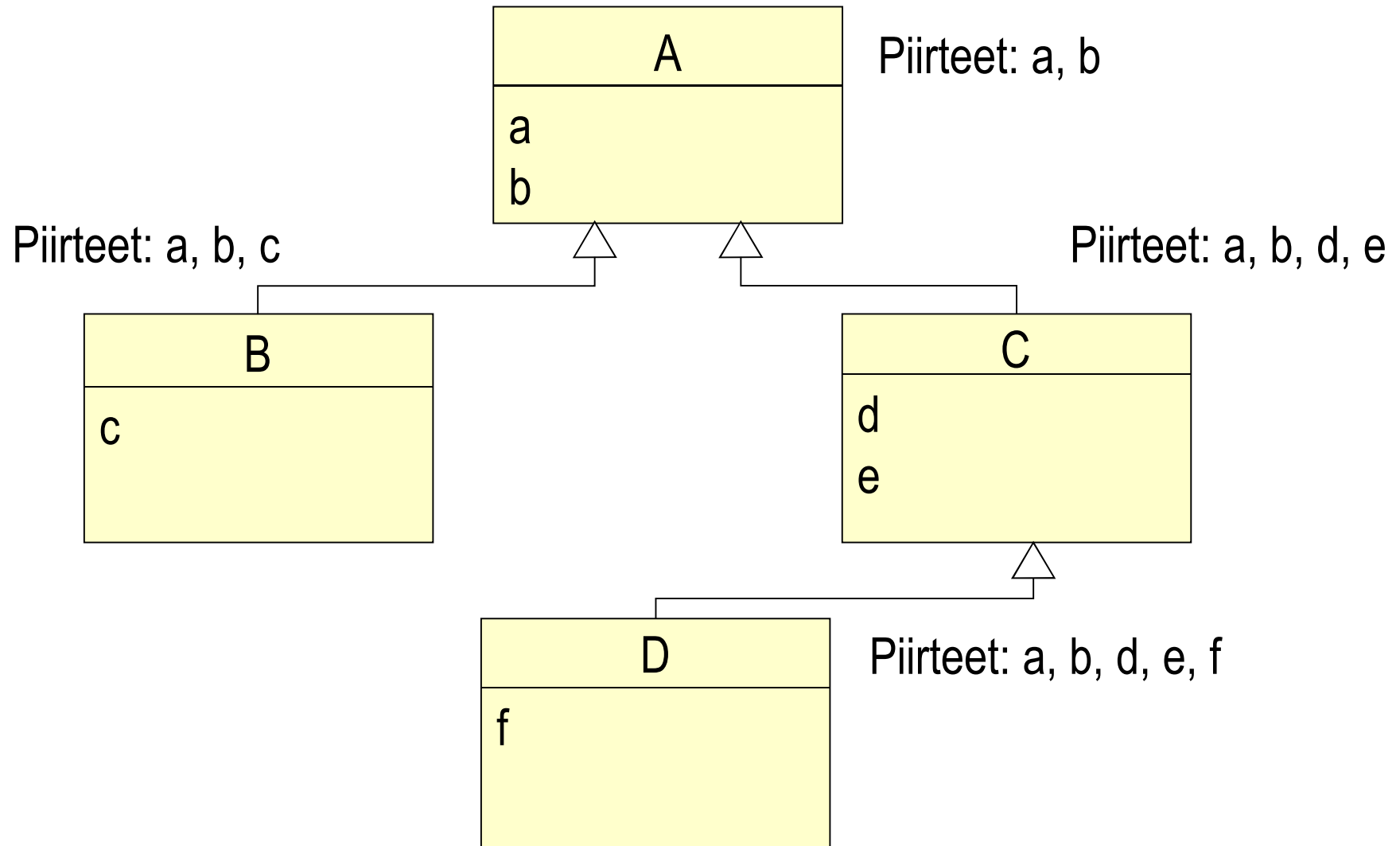
- UML:n luokkakaaviossa luokkaa symboloi laatikko, jossa annetaan luokan nimi ja usein myös luokan piirteet. Attribuutit ja metodit erotetaan myöhemmin väliviivalla, kun luokkasymboleita ryhdytään piirtämään hieman tarkemmin. Periytymissuhde piirretään onttokärkisenä yliluokkaan osoittavana nuolena.

Mitä on periytyminen?

- Yliluokalla on yksi tai useampia aliluokkia.
- Aliluokka tuntee yliluokkansa, mutta ei aliluokkia.
- Luokka *A* on luokan *B* **esivanhempi** (ancestor) jos *A* on *B*:n yliluokka tai *B*:n yliluokan esivanhempi.
- Luokka *B* on luokan *A* **jälkeläinen** (descendant), jos *B* on *A*:n aliluokka tai *A*:n jonkin aliluokan jälkeläinen.

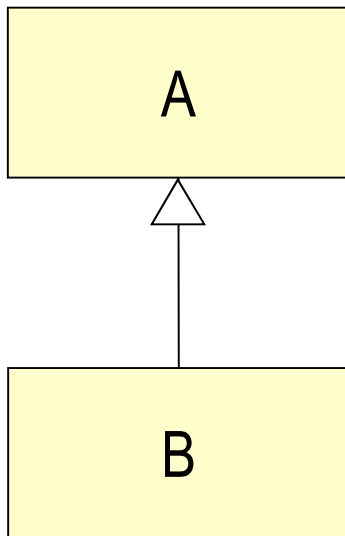


Mitä on periytyminen?

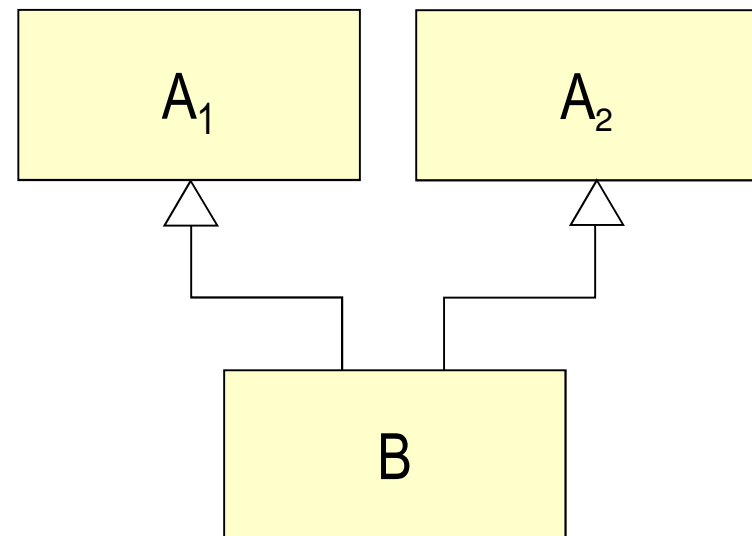


Yksittäis- ja moniperiytyminen

- **Yksittäisperiytymisessä**
(single inheritance)
jokaisella aliluokalla on
yksi yliluokka.

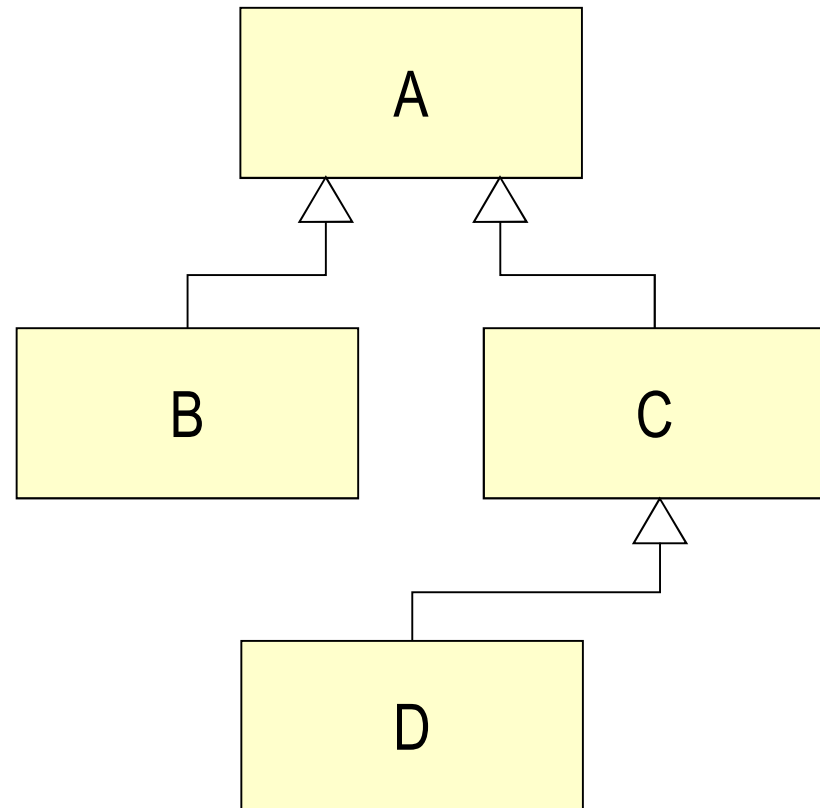


- **Moniperiytymisessä**
(multiple inheritance)
aliluokalla voi olla
useampia yliluokkia.



Yksittäis- ja moniperiytyminen

- Yksittäisperiytyminen tuottaa selkeän puumaisen rakenteen.
- Moniperiytyminen on kuvattavissa vaikeampiselkoisena verkkona.
- Molemmiin tavoin saadaan aikaiseksi luokkahierarkia, jolla on **juuri** (root) tai juuria.

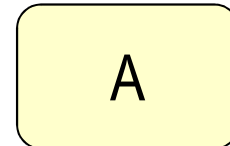


Yksittäis- ja moniperiytyminen

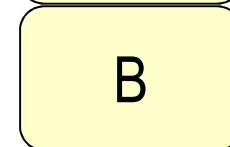
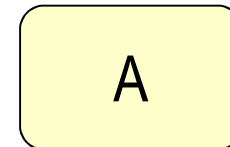
- Moniperiytymisestä sekä hyötyä että haittaa.
- Joissakin kielissä, Java mukaan lukien, on vain yksittäisperiytyminen.
 - Moniperiytyminen voidaan ilmaista Javassa rajoitetusti rajapintojen avulla.
- Python ja C++ ovat kieliä, jossa luokalla voi olla useita yliluokkia.
- Jatkossa periytymisen esittely rajoitetaan vain yksittäisperiytymiseen.

Oliot ja perityt luokat

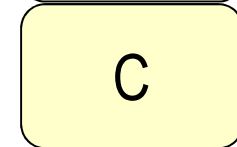
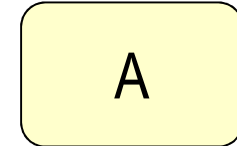
- Periytymisen kautta oliolla voi olla **useampia** luokkia.
- Jatkossa olion **perus-luokaksi** sanotaan luokkaa, josta olio on luotu.
- Kukin yliluokka lisää olioon uuden kerroksen piirteitä (attribuutteja ja/tai metodeja).



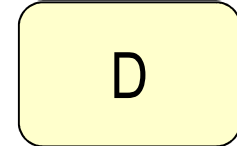
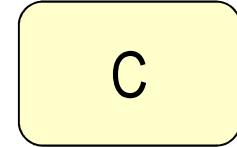
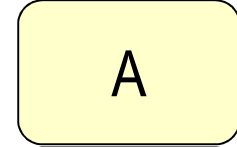
A-luokan
olio



B-luokan
olio



C-luokan
olio



D-luokan
olio

Periytymisen käytännön näkökulmasta

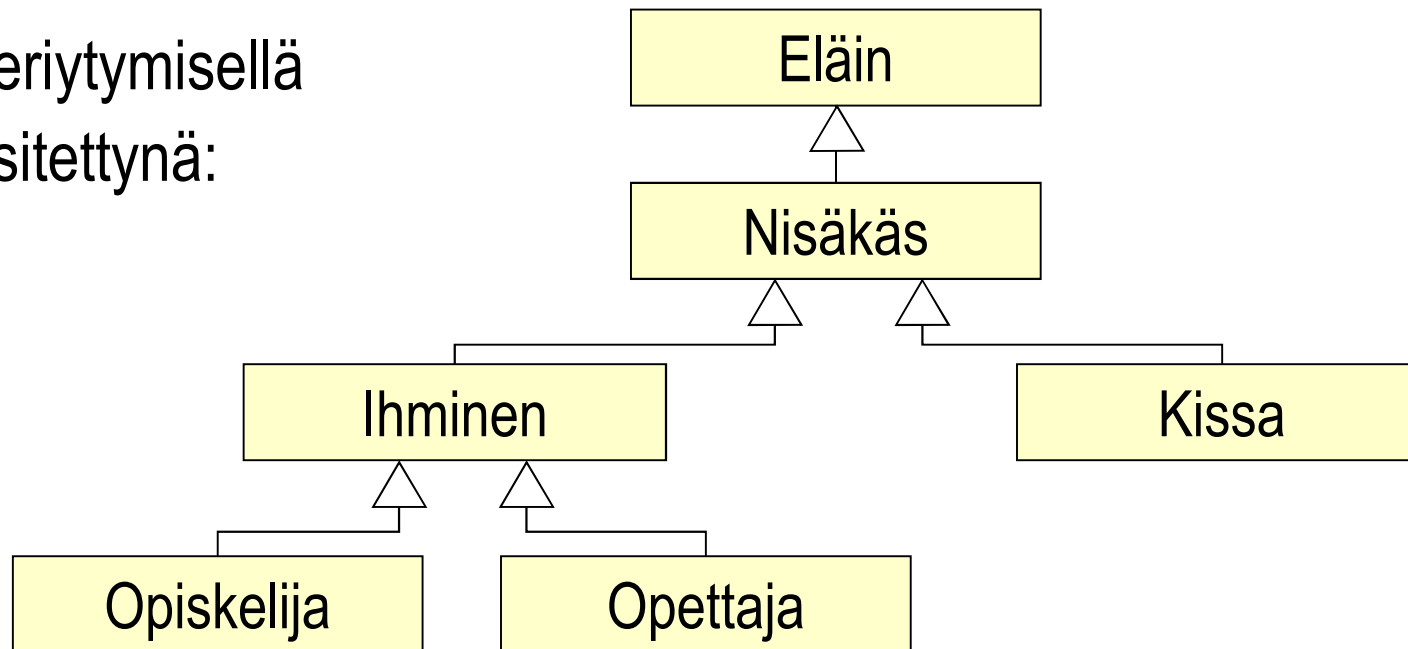
- Periytyminen tukee koodin uudelleenkäyttöä; aliluokkien ei tarvitse toistaa perittyjä piirteitä.
- Periytymisen avulla voidaan kehittää ohjelmistoja tarkentaen (inkrementaalisesti), koska aliluokkia voidaan lisätä ilman, että yliluokkia tarvitsee muuttaa.
- Periytymisen avulla voidaan toteuttaa monimuotoisuutta.

Periytyminen filosofisesta näkökulmasta

- Periytyminen tukee käsitteelliseen mallintamiseen perustuvaa ohjelmistokehitystä: aliluokka on käsitteen erikoistus ja yliluokka on yleistys.
- Periytymisellä voidaan mallintaa käsitehierarkioita.
- Periytymistä kutsutaankin joskus **yleistämiseksi** (generalization), koska periytyminen liittyy läheisesti käsitteellisestä mallintamisesta tuttuun IsA-suhteeseen.
 - Periytyminen on luontevaa, jos lause “*Aliluokka IsA Yliluokka.*” tuntuu järkevältä. Esimerkiksi “*Auto IsA Ajoneuvo.*”

Periytymisen soveltaminen

- Eräs käsitehierarkia periytymisellä esitettynä:

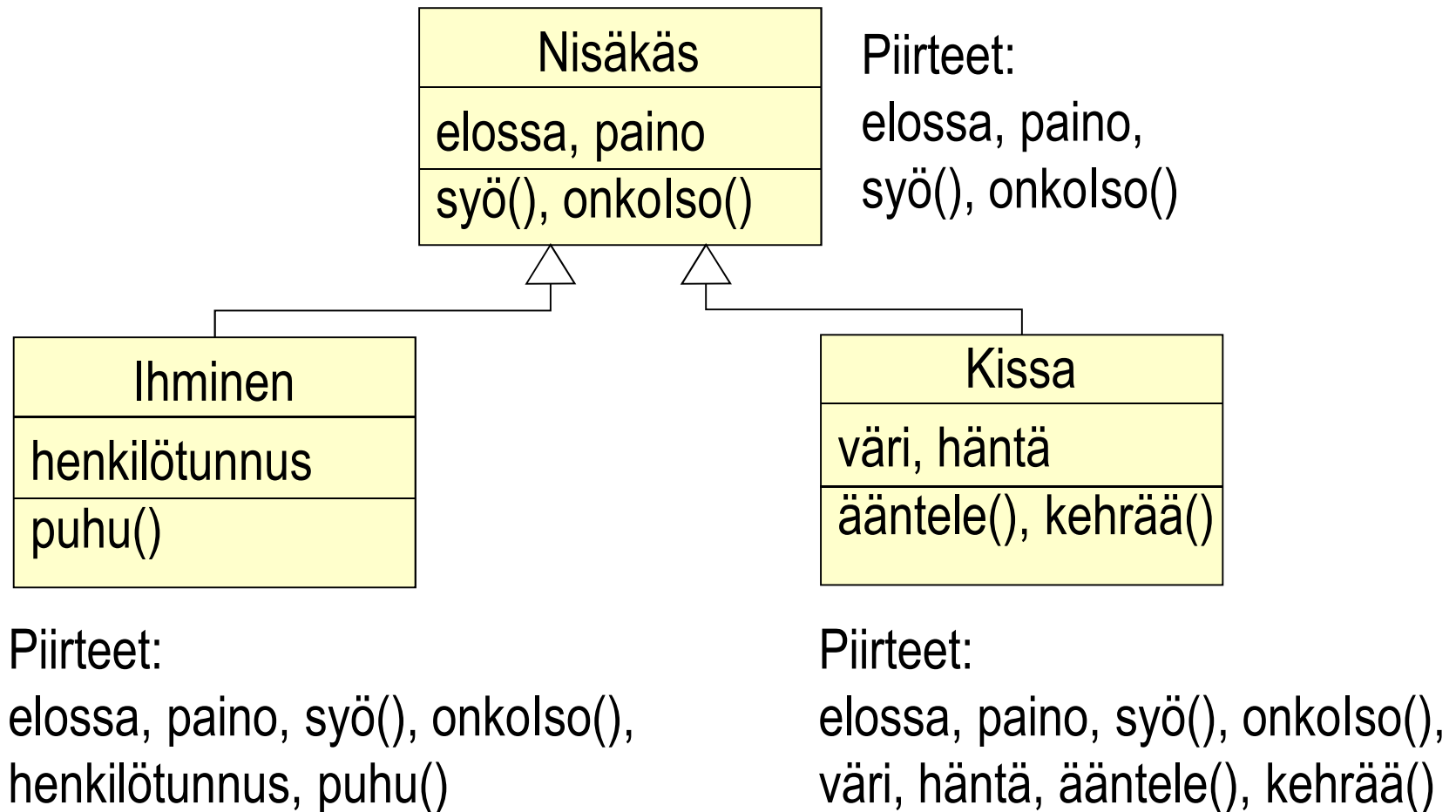


- *Nisäkäs* IsA *Eläin*.
- *Ihminen* IsA *Nisäkäs*, *Kissa* IsA *Nisäkäs*.
- *Opiskelija* IsA *Ihminen*, *Opettaja* IsA *Ihminen*.

Periytymisen soveltaminen

- Aliluokan tulisi aina tuoda ylliluokkaansa jokin tarkennus (tai laajennus).
 - Aliluokassa voidaan määritellä uusia piirteitä (attribuutteja ja metodeja).
 - Aliluokka voi **korvata** (override) ylliluokassa ilmentymämetodilla määritellyn toiminnallisuuden toteuttamalla perityn metodin eri tavoin. Korvaaminen on eräs monimuotoisuuden muoto.
 - Aliluokka tyypillisesti rakentaa olion omalla tavallaan.

Nisäkäs, kissa ja ihminen



Kissa-olio

- *Misse*-olio luodaan *Kissa*-luokasta: *Kissa* on *missen* perusluokka.
- Koska jokainen kissa on nisäkäs, on *missen* on sekä *Kissa* että *Nisäkäs*.
- *Misse* perii *Nisäkäs*-luokasta attribuutit *elossa* ja *paino* sekä metodit *syö()* ja *onkolso()*, jotka voidaan korvata *Kissa*-luokassa siten, että ne toimivat kissalle ominaisesti.
- Lisäksi *missellä* on kissoille ominaiset piirteet (esim. attribuutti *häntä* ja metodi *kehrää()*).

elossa, paino
syö(), *onkolso()*

Nisäkäs-kerros

väri, häntä
ääntele(), *kehrää()*

Kissa-kerros