

Ohjelmoinnin peruskurssi Y1

CS-A1111

Debuggaus

Oppimistavoitteet: tämän videon jälkeen

- ▶ Tiedät, miten ohjelman toimintaa voi tutkia ja ohjelmassa olevia virheitä etsiä debuggerin avulla.

Debuggeri

- ▶ Tyypillinen tilanne: ohjelma on kirjoitettu, Python-tulkki ei valita virheistä, mutta ohjelma toimii jotenkin väärin.
- ▶ Ohjelman toimintaa voi tutkia ja virhettä yrittää etsiä *debuggerin* avulla.
- ▶ Debuggerin avulla ohjelmaa voi ajaa käsky kerrallaan ja tutkia samalla ohjelman käyttämien muuttujien arvoja.
- ▶ Ohjelmaan voi asettaa pysähdyspisteitä (breakpoint), joissa ohjelma pysähtyy.
- ▶ PyCharmiin kuuluu debuggeri, jonka avulla voi ajaa mm. Python-ohjelmia.

Debuggerin käyttö PyCharmissa

- ▶ Ennen debuggerin käynnistymistä kannattaa asettaa ensimmäinen pysähdyspiste. Valitaan haluttu rivi koodista, painetaan rivin vasemmassa reunassa hiiren vasenta painiketta – reunaan ilmaantuu punainen pallo.
- ▶ Tämän jälkeen debuggerin voi käynnistää Run-valikosta tai valitsemalla hiiren oikealla painikkeella avautuvasta valikosta `Debug 'tiedoston_nimi'`
- ▶ PyCharm ajaa ohjelmaa ensimmäiseen pysähdyspisteeseen saakka ja siirtyy Debug-perspektiiviin (ellei se ole jo käytössä).
- ▶ Tämän jälkeen ohjelmassa voi edetä rivi kerrallaan valitsemalla `Run->Step Into`.

Debuggerin käyttö PyCharmissa, jatkoa

- ▶ Jos rivillä on funktion kutsu, ja haluaa siirtyä funktion sisälle, on valittava Step Into. Jos haluaa siirtyä pois funktiosta takaisin sitä kutsuneeseen kohtaan, valitaan Step Out. Jos ei haluta siirtyä funktion sisälle, voi valita Step Over.
- ▶ Muuttujien arvot näkyvät joka vaiheessa ikkunan alaosassa.
- ▶ Ideana on ajaa ohjelmaa käsky kerrallaan ja katsoa, miten muuttujien arvot muuttuvat.
- ▶ Kun jossain vaiheessa arvo ei vastaa odotettua tai siirrytään eri riville kuin pitäisi, voi miettiä tarkemmin, miksi näin tapahtuu. Usein tämä johtaa virheen jäljille.
- ▶ Debuggauksesta pääsee takaisin ohjelman editointiin sulkemalla Debug-ikkunan vasemmassa reunassa olevasta punaisesta raksista.