

# Debuggaus ja debuggeri

# Debuggaus

- Virheiden *paikallistamista* ja *poistamista*
- Jotta mahdollista, virheen olemassaolo täytyy ensin tiedostaa (testaus)
- Usein virheen paikallistaminen on työläin vaihe
- Mitä aiemmin virhe havaitaan, sitä helpompaa paikallistaminen on
- Mitä parempi testiaineisto, sitä helpompaa virheen paikallistaminen on

# Paikallistaminen 1: testitulostukset

- Ehkä yksinkertaisin tapa virheen paikallistamiseen:
- Lisätään oletetun virhekohdan ympärille testitulostuksia / assert-testejä, jotka
  - näyttävät/testaavat ohjelman kulun
  - näyttävät/testaavat muuttujien arvoja
- Ongelma: työläs, vaatii koodin muokkaamista jatkuvasti

# Paikallistaminen 2: debuggeri

- Debuggeri: ohjelma, jolla ohjelman kulkua ja muuttujia voi seurata (ja vaikuttaa niihin)
- Usein ohjelmointikielikohtaisia, vaihtelevat kyvykkyydeltään
- Vaativat usein tukea kääntäjältä (kääntäjä tuottaa debuggerin vaatimat lisätietorakenteet)
- Kääntäjän optimoinnit voivat sekoittaa debuggeria (suoritettu koodi ei vastaa kirjoitettua)
- Ohjelman voi suoraan ajaa debuggerissa tai käynnistää debuggerin kaatumisen jälkeen

# Debuggeritermejä

- **keskeytyskohta (breakpoint)**: kohta ohjelmassa, johon suorituksen halutaan pysähtyvän
- **askellus (stepping)**: ohjelman suorittaminen vähittäin rivi tai funktiokutsu kerrallaan (*step over, step into, step out*)
- **tarkkailukohta (watchpoint)**: näkymä halutun muuttujan tai muistialueen sisältöön
- **kutsujälki (stack trace)**: Funktioiden kutsuhierarkia ohjelman pysähdyttyä

# Tyypillinen debuggaustilanne

1. Selvitetään vikapaikka
  - Testeistä
  - Kaatumisen jälkeen debuggerilla (**kutsujälki**)
2. Asetetaan **keskeytyskohta** *ennen* oletettua virhekohtaa
3. **Askelletaan** ohjelmaa, **tarkkaillaan** muuttujien arvoja, funktioiden parametreja ja paluuarvoja jne. kunnes virhe löytyy
4. Jos mentiin ohi, tarkennetaan hakua ja yritetään uudelleen