

DAT171 Object oriented programming in Python

- ▶ 3 Inlämningsuppgifter (3.0hp)
- ▶ Tentamen i datorsal (4.5hp)
- ▶ Förkunskapskrav: Grundläggande kunskap i programmering, såsom variabler, villkor, loopar, och funktioner i något programmeringsspråk.
- ▶ Kurslitteratur: Föreläsningsanteckningar, onlinematerial, frivillig bok "Python for everyone"
- ▶ 2 föreläsningar / vecka. Ges på Engelska
- ▶ 3 handledda datorlabbstillfällen / vecka
- ▶ Lärare
 - ▶ Thomas Svedberg, thomas.svedberg@chalmers.se, C3SE (examinator)
 - ▶ Mikael Öhman, mikael.ohman@chalmers.se, C3SE
 - ▶ Hugo Strand, hugo.strand@chalmers.se, C3SE

Varför Python

- ▶ Kan köras interaktivt, enkel syntax, enkel debugging
- ▶ Många programmeringstekniker; objektoreinterad, funktionell, imperativ
- ▶ Används som scriptspråk i många applikationer (e.g. Abaqus, Salome, Autodesk)
- ▶ Många bibliotek skrivna i andra språk har bindningar till Python (Qt, OpenFOAM, TensorFlow, ...)
- ▶ Alumnenkät frågar efter mer programmering
- ▶ Volvo frågar efter python-programmerare

Varför Objektorienterad programmering

- ▶ Kraftfull programmeringsparadigm med brett användande i industri och open source.
- ▶ Låter oss bygga nya datastrukturer, som höjer abstraktionsnivån av programmet genom att skapa betydelse av rå, underliggande, data.
- ▶ Man kombinerar data och funktioner för att skapa objekt.
- ▶ Du har förmodligen redan använt objekt i MATLAB; Matriser, textsträngar, filer, ...
- ▶ Exempel på objekt man skapar/använder är brett; grafiska knappar, bilder, regex-sökning, datum och tider, enheter, koordinater, funktioner.

Kursmål

- ▶ *Lärandemål: Efter avklarad kurs skall studenten självständigt kunna skriva objektorienterad mjukvara med Python. Vidare skall studenten kunna läsa referenslitteratur för programmeringsspråket och speciellt kunna använda Scipy/Numpy för numeriska beräkningar, och PyQt för att skriva grafiska användargränssnitt.*
- ▶ **Innehåll:**
 - ▶ Grundläggande byggstenarna i Python (variabler, villkor, loopar, bibliotek, felsökning)
 - ▶ Datastrukturer (träd, ordböcker, köer, tupler)
 - ▶ Objektorienterad programmering (klasser, objekt, arv, polymorfism, abstrakta klasser)
 - ▶ Qt för att skapa grafiska användargränssnitt
 - ▶ NumPy (matriser, vektorer, linjär algebra)
 - ▶ Scipy (paker för numeriska beräkningar)
 - ▶ Matplotlib
 - ▶ IPython (interactive python)

Examination

- ▶ 3 obligatoriska inlämningsuppgifter i grupp om 2 där ni kommer att få:
 1. använda IPython + Numpy + Matplotlib ~ Matlab-ersättare
 2. skriva egna datastrukturer med hjälp av klasser, och göra ett eget bibliotek
 3. göra interaktiva program med grafiska användargränssnitt
- ▶ Tentamen som betygsätter kursen
 - ▶ ges i datorsal
 - ▶ består typiskt av 4 uppgifter (7p filer och text, 5p numpy/scipy, 6p klasser, 7p klasshierarkier med arv)
 - ▶ 10p = 3:a, 15p = 4:a, 20p = 5:a
 - ▶ tillåter all kurslitteratur, inkl föreläsningsanteckningar + allt referensmaterial för python, numpy, scipy offline