

Ohjelmoinnin peruskurssi Y1

CS-A1111

Ohjelmointiprojektin vaiheet ja funktion suunnittelu

Oppimistavoitteet: tämän videon jälkeen

- ▶ Tiedät, millaisia vaiheita ohjelman suunnittelussa ja kirjoittamisessa on sekä
- ▶ osaat käyttää niitä periaatteita, joiden mukaan ohjelman rakennetta voi suunnitella.

Ohjelmointiprojektin vaiheet

1. Määrittely

- ▶ Mitä ohjelma täsmällisesti ottaen tekee?
- ▶ Miten ohjelma kommunikoi käyttäjän kanssa?

2. Ohjelman suunnittelu (ohjelman rakenne ja ohjelman käyttämät tietorakenteet)

- ▶ Mitä funktioita (tai muita laajempia osia) ohjelmassa on? Mitä parametreja funktioilla on ja millaisia arvoja ne palauttavat?
- ▶ Mitä tietorakenteita (esim. listat, sanakirja jne) ohjelma käyttää?

3. Koodaus ohjelmointikielelle

4. Testaus (rinnakkain koodauksen kanssa)

- ▶ Testataan aluksi mahdollisimman pieniä osia (esim. yksittäisiä funktioita) kerrallaan, sitten laajempia kokonaisuuksia.

5. Käyttöönotto

6. Ylläpito

- ▶ Mahdollisten virheiden korjaus ja uusien ominaisuuksien lisääminen.

Suunnittelu: mitä funktioita ja tietorakenteita ohjelmaan tulee?

- ▶ Kirjoita kuvaus ohjelman toiminnasta.
- ▶ Millaisista osatehtävistä ohjelman toiminta koostuu?
- ▶ Yleensä kutakin osatehtävää varten kirjoitetaan oma funktio.
- ▶ Aloita ohjelman tärkeimmistä osatehtävistä ja tarkenna sitten näiden toimintaa. Tällöin saattaa osoittautua tarpeelliseksi määritellä uusia osatehtäviä.
- ▶ Lisäksi on mietittävä, mitä tietoja funktio tarvitsee muulta ohjelmalta (parametrit) ja mitä tietoja se tuottaa muulle ohjelmalle (paluuarvot).
- ▶ Huomaa: tämä lähestymistapa ei sovi olio-ohjelmointiin.
- ▶ Tietorakenteet: Mieti, mitä tietoa ohjelma joutuu käsittelemään ja missä muodossa se kannattaa tallentaa. Tarvitaanko esim. merkkijonoja, listoja, sanakirjoja tms. yksittäisiä lukuja esittävien muuttujien lisäksi?

Lisää funktioiden suunnittelusta

- ▶ Tavoitteena on se, että funktio näyttää ulkopuolelle mustalta laatikolta: funktion käyttäjän tarvitsee tietää, mitä lähtötietoja funktio tarvitsee ja mitä se palauttaa, mutta ei funktion toiminnan yksityiskohtia.
- ▶ Funktion sisäinen toteutus ei saa vaikuttaa muuhun ohjelmaan.
- ▶ Funktioiden pituus pitää suunnitella sopivaksi. Yhden rivin mittaisista käskyistä kannattaa yleensä tehdä funktioita vain silloin, jos ne laskevat jonkin matemaattisen lausekkeen arvon tai tarkastavat monimutkaisemman ehdon totuusarvon. Toisaalta liian pitkät funktiot vaikeuttavat ohjelman rakenteen ymmärtämistä.
- ▶ Funktion pitäisi olla loogisesti yhtenäinen kokonaisuus.

Apufunktio luvun lukemiseen

- ▶ Jos samassa ohjelmassa luetaan kokonaislukuja useassa kohdassa, kannattaa kirjoittaa apufunktio kokonaisluvun lukemiseen.
- ▶ Funktio lukee ja palauttaa kokonaisluvun. Se pyytää käyttäjältä uutta kokonaislukua niin kauan, että saadaan kelvollinen kokonaisluku.
- ▶ Vastaavat apufunktiot voidaan kirjoittaa myös muuntyyppisten arvojen lukemiseen.

Kokonaisluvun lukeminen apufunktion avulla: koodi

```
def lue_kokonaisluku():
    luku_onnistui = False
    while not luku_onnistui:
        try:
            luku = int(input())
            luku_onnistui = True
        except ValueError:
            print("Virheellinen kokonaisluku!")
            print("Anna uusi!")
    return luku
```


Kokonaisluvun lukeminen apufunktion avulla: koodi jatkuu

```
def main():
    NAULAKERROIN = 0.4536
    print("Muutan nauloina annetun massa kilogrammoiksi.")
    print("Anna massa nauloina.")
    naulat = lue_kokonaisluku()
    kilot = NAULAKERROIN * naulat
    print("Massa on {:.3f} kg".format(kilot))

main()
```