

Autotallin ovi - Tehtävänanto

Sisälllys

Pisteytys.....	1
Tehtävä:	1
Testit:	2
Ohjeet:	4
Reflektointi	9

Pisteytys

- Max 9p
- 1 piste per läpi mennyt testi (max 7p)
- Reflektointitehtävä (max 2p – yksi piste jos osaa vastata edes osittain ja toinen piste tyhjentävästä vastauksesta)

Tehtävä:

Tässä harjoituksessa ohjelmoidaan autotallin ovi, joka toimii seuraavilla toimintaperiaatteilla:

1. Ovi lähtee liikkeelle napin painalluksesta
2. Ovi pysähtyy joko napin painalluksesta tai saavuttaessaan ääriasennon
3. Ovi lähtee liikkeelle aina eri suuntaan kuin mitä se liikkui ennen pysähdystä. Esim. jos ovi liikkui alaspäin ennen pysähdystä, se lähtee liikkeelle ylöspäin.

Näin ollen, valmis toteutus pitäisi läpäistä seuraavat testitapaukset (jokaisesta testitapauksesta saa yhden pisteen).

Testit:

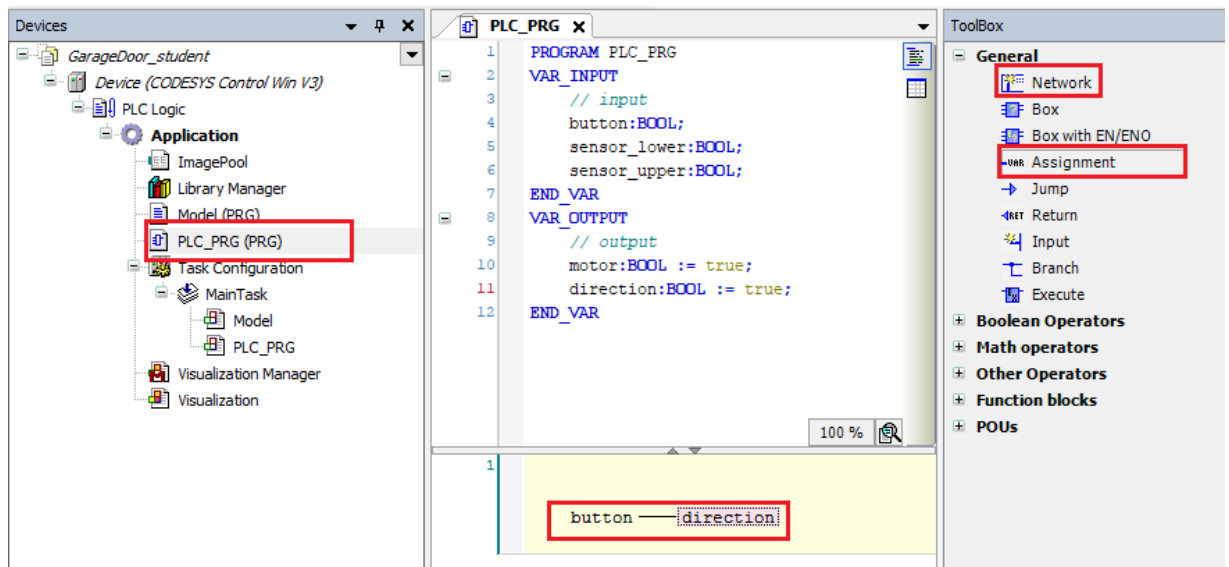
Testi#	Alkutila	Syöte	Odotettu lopputulos
1	Ovi ylhäällä	Remote nappia painetaan	Ovi liikkuu (moottori päällä) alas kunnes Remote nappia painetaan tai saavutaan ala-asentoon
2	Ovi liikkuu alas	Remote nappia painetaan	Liike pysähtyy
3	Ovi pysäytetty matkalla alas	Remote nappia painetaan	Ovi liikkuu (moottori päällä) ylös kunnes Remote nappia painetaan tai saavutaan yläasentoon
4	Ovi alhaalla	Remote nappia painetaan	Ovi liikkuu (moottori päällä) ylös kunnes Remote nappia painetaan tai saavutaan yläasentoon
5	Ovi liikkuu ylös	Remote nappia painetaan	Liike pysähtyy
6	Ovi pysäytetty matkalla ylös	Remote nappia painetaan	Ovi liikkuu (moottori päällä) alas kunnes Remote nappia painetaan tai saavutaan ala-asentoon
7	Ovi liikkuu	Remote nappia painetaan ja pidetään pohjassa.	Ovi pysähtyy, eikä sen tila vaihtele liikkumisen ja pysähtymisen välillä (eli signaalin pulssimainen luonne on otettu huomioon)

Autotallin oven voi toteuttaa monella eri tavalla eri PLC-kielillä (esim. ST, FBD, SFC jne.)

Alla on esitelty miten ohjelma toteutetaan FBD-kielillä.

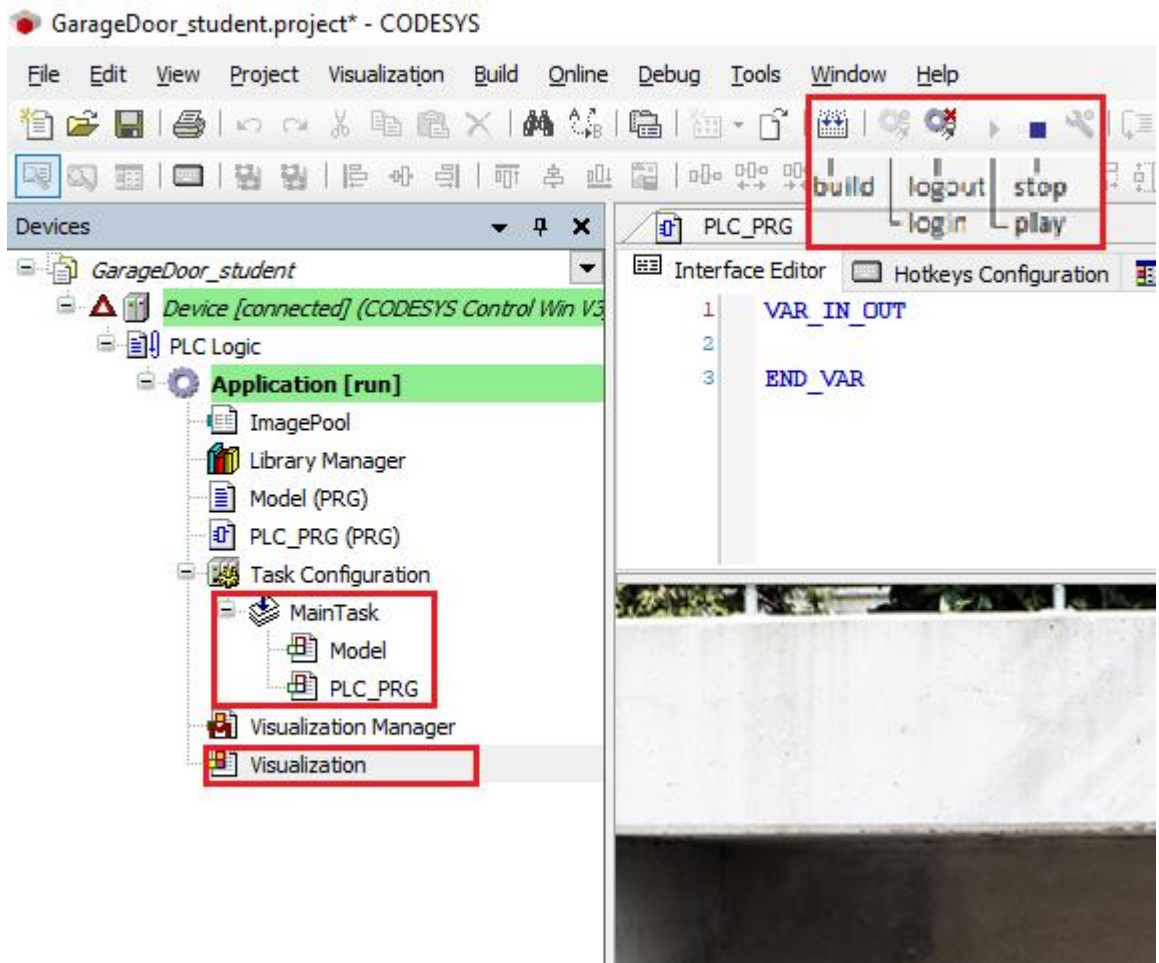
Ohjeet:

1. Lataa MyCourses sivulta autotallin oven projektipohja ja avaa se Codesys-ohjelmalla
2. Erittäin todennäköisesti käyttämäsi ohjelmistoversio on uudempi, kuin MyCourses sivuilta saatava pohja, jolloin ohjelmisto herjaa puuttuvista kirjastoista. Ota esille MyCourses sivulta löytyvä ” Ohjeita Codesys ongelmatilanteisiin” ja toimi sen ohjeiden mukaan.
3. Aloita autotallin oven FBD-logiikan ohjelmoiminti tuplaklikkaamalla projektin tiedosto PLC_PRG auki ja vetämällä FBD-elementtejä oikealla olevasta ToolBox-paneelistä. Mikäli PLC_PRG ei aukea, kirjautu virtuaali PLC:lle sisään ja avaa tuplaklikkaa PLC_PRG:tä. Jos PLC_PRG aukeaa, kirjautu ulos.
4. Voit aloittaa vetämällä network-elementin editoriin ToolBoxista. Sen jälkeen vedä assignment-elementin editoriin ja laita sen muuttujiksi button ja direction (ks. kuva 2).



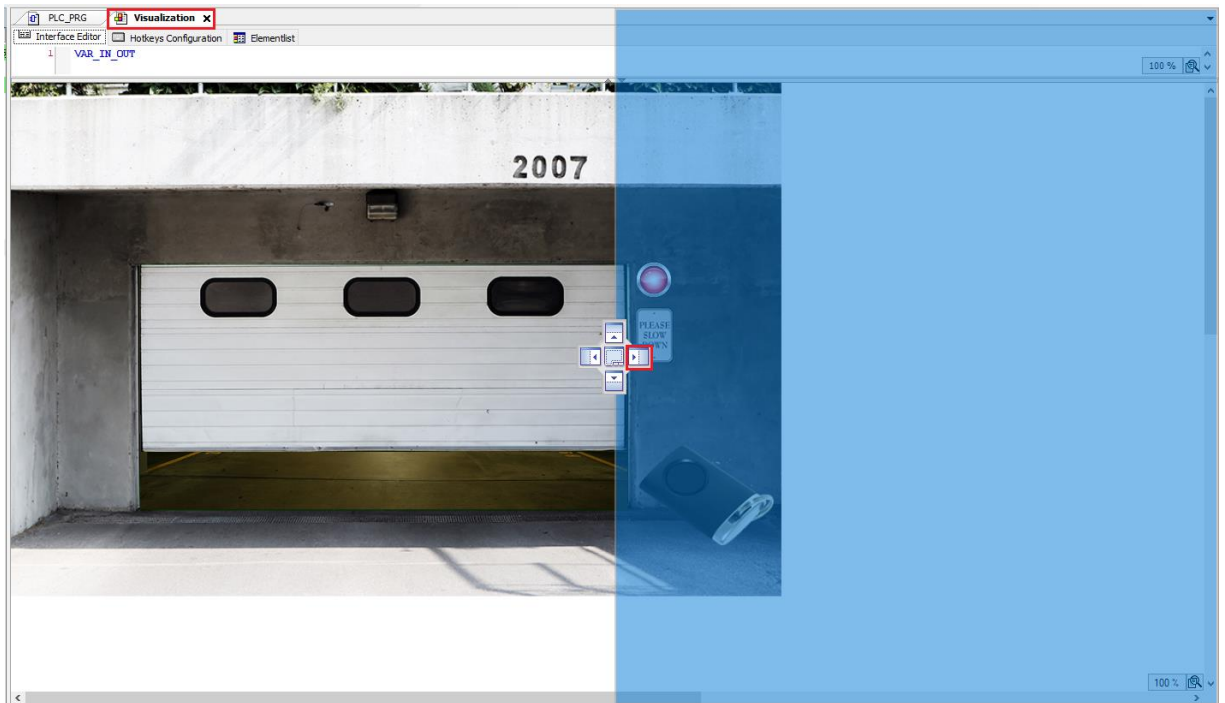
Kuva 1: FBD-editorin käyttö

5. Moottori on alustettu olemaan aina päällä PLC_PRG:n muuttujissa. Autotallin toiminta on mallinnettu Model(PRGR):ssä. Lähtökohtaisesti älä missään tilanteessa muuta Modelia tässä tai muissa tehtävissä. Tehtävät on laadittu niin, että opiskelijan ei käytännössä tarvitse koskea Modeliin. Model(PRGR):stä nähdään, että ovi on määritetty kulkemaan alaspäin, kun directionin arvo on 1. Muuttamalla buttonin arvoa, saadaan nyt suunta muuttumaan. Tarkasta, että PLC_PRG ja Model ovat MainTaskissa. Kokeile ohjelmaa kirjautumalla virtuaaliselle PLC:lle, suorittamalla kylmä resetoinnin ja käynnistämällä ohjelman (kuva 2). Siirry Visualization välilehdelle ja paina painiketta pohjassa.

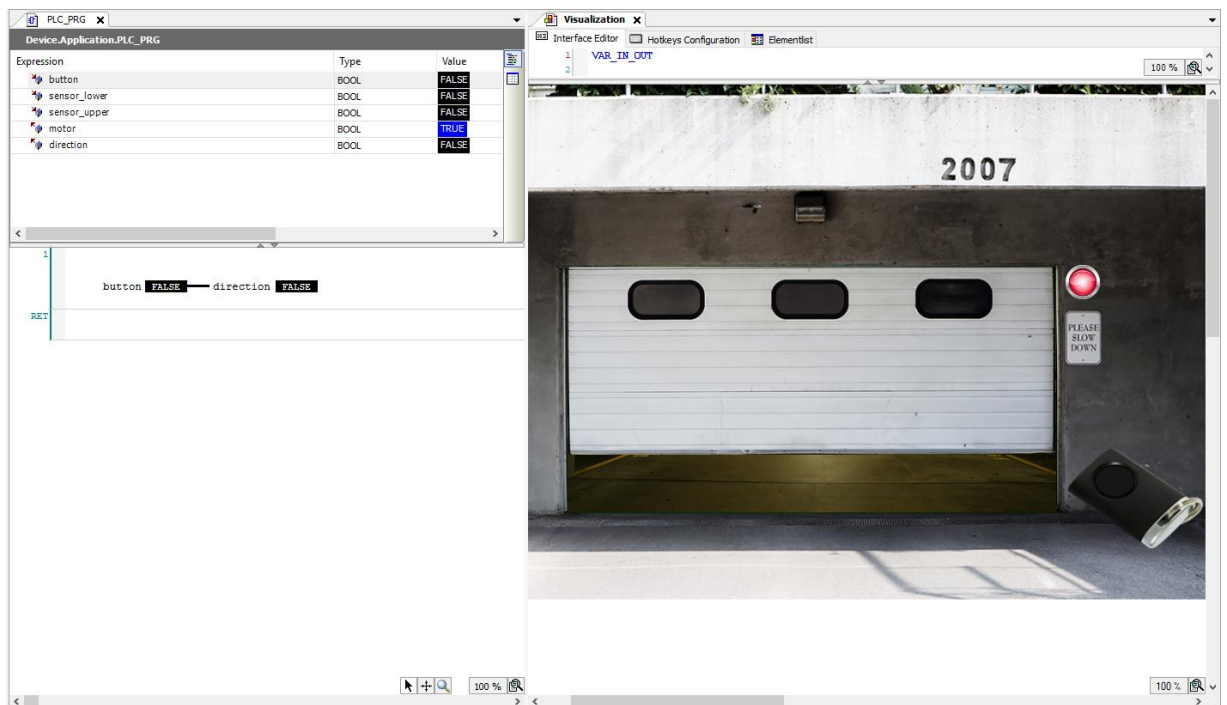


Kuva 2: Ohjelman testaus

6. Testataksien ohjelman toimivuutta, voi olla hyvä tarkastella inputtien ja outputtien arvoja simulaation ollessa käynnissä. Simulaation ollessa käynnissä PLC_PRG välilehdellä näkyy reaaliaikaiset IO- tiedot. Ota kiinni Visualization välilehdeltä ja raahaa se näyttöön tulevan ristikon oikean nuolen päälle (kuva 3). Tällöin näyttö jakautuu kahteen osaan ja voit tarkastella sekä visualization välilehteä, että PLC_PRG:tä (kuva 4).



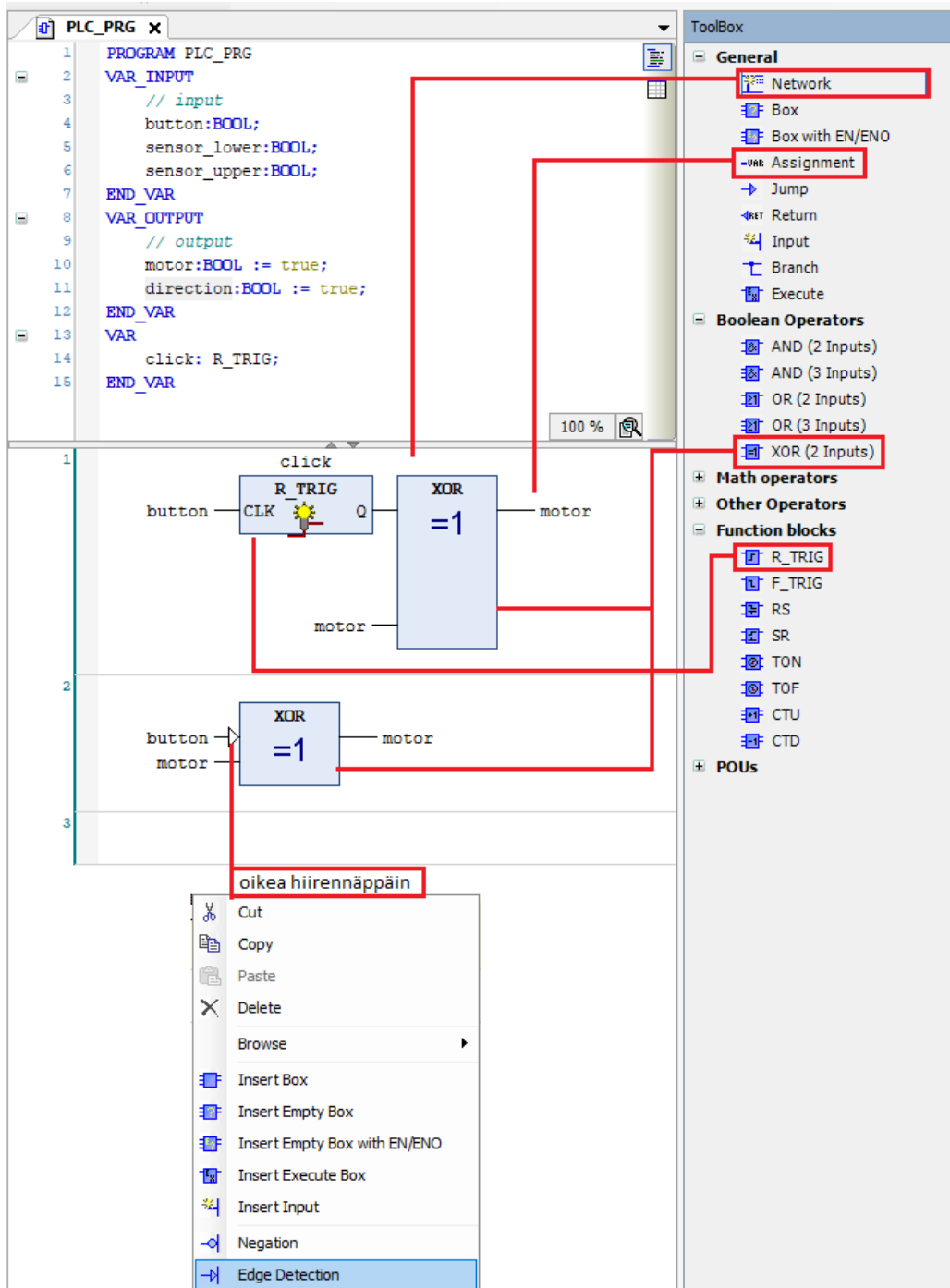
Kuva 3: Ruudun jako kahteen



Kuva 4: Kaksi näkymää - PLC_PRG ja Visualization

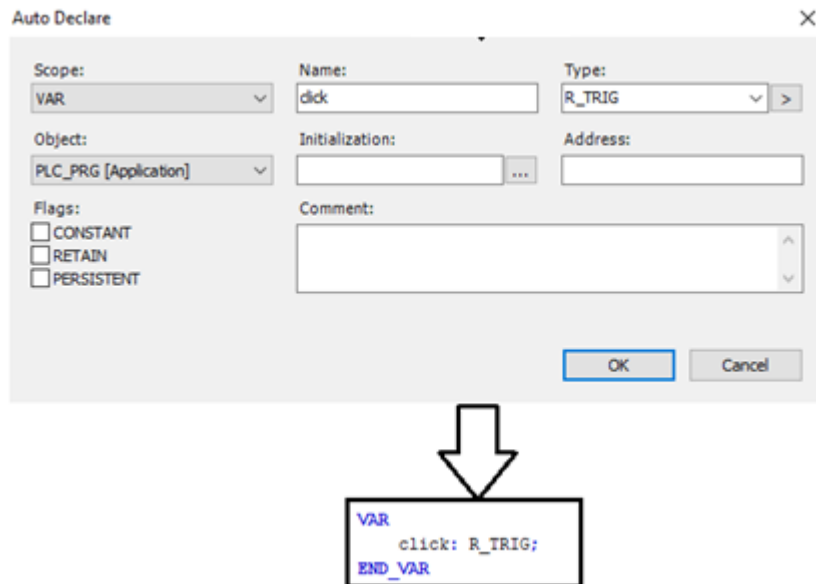
7. Tekemämme ratkaisu ei sellaisenaan vastaa yhteenkään testivaatimukseen. Visuaalisen tarkastelun perusteella voidaan havaita, että järjestelmässä on viivettä. Oikeastaan kyse on siitä, että painaessa nappia, nappi jää helposti pohjaan monen ohjelmasyklin ajaksi, eikä ohjelma siten toimi tarkoituksenmukaisesti. Kirjautu

virtuaaliselta PLC:ltä ulos. Järjestelmän Malli (model) ja lokaalit muuttujat on alustettu sellaisiksi, että moottori on jatkuvasti päällä ja se pyrkii alas. Luodaan ratkaisu, joka pysäyttää moottorin napin painalluksesta ja käynnistää moottorin, kun nappia painetaan uudestaan. Tämän toiminnallisuuden voi toteuttaa esimerkiksi käyttämällä XOR- tai SEL-toimilohkoa ja R_TRIG tai Edge Detectonia. Toimilohkojen kuvauksen voit katsoa FBD-kielen oppaasta. Alla on kaksi esimerkkiä, jolla saat moottorin muuttamaan tilaansa napin painalluksesta (kuva 5).



Kuva 5: Moottorin tilan muuttaminen napin painalluksella

8. Mikäli käytät R_TRIG- lohkoa (esimerkki 1), pyytää Codesys ensimmäisenä muuttamaan instanssin nimeä. Kun nimen muuttaa kuvaavaksi (tässä tapauksessa "click" on oikein hyvä) ja painaa Enteriä, niin Codesys avaa Auto Declare ikkunan. Tarkasta nimi ja tyyppi ja paina OK. Nyt lokaaleissa muuttujissa pitäisi olla VAR määrittelyn alla click niminen, R_TRIG – tyyppin muuttuja (kuva 6).



Kuva 6: R_TRIG - tyyppisen muuttujan määrittely

9. Seuraava vaihe on tehdä ohjelma loppuun tehtävänannon mukaan.

Reflektointi

Ohjelman pystyy toteuttamaan yhdelle riville tai monelle. Jos toteutit ohjelman yhdelle riville pohdi miten voisit jakaa sen monelle. Jos taas toteutit ohjelman monelle riville pohdi miten voisit tehdä sen yhdellä rivillä.

Pohdi kumpi toteutuksista on yksinkertaisempi ja helpompi.

”Yksinkertaisuus” ja ”helppous” ovat hieman epämääräisiä käsitteitä, mutta ne liittyvät läheisesti sellaisiin ohjelmiston laatuominaisuuksiin kuten ylläpidettävyyteen ja skaalautuvuuteen, joihin mennään seuraavassa harjoituksessa.