

Johdatus ohjelmointiin

Syksy 2006
Viikko 5
4.10. - 5.10.

1

matti.tedre@cs.joensuu.fi



Tällä viikolla käsiteltävät asiat

- Loogiset operaattorit
 - **and** , **or**
- Sisäkkäiset ohjelmarakenteet
 - Sisäkkäiset toistolauseet
 - Sisäkkäiset ehtolauseet
 - Muut vastaavat
- Tyypillisiä virhetilanteita ?
 - ...Toistolauseissa
 - ...Ehtolauseissa

2

matti.tedre@cs.joensuu.fi



Vertailujen yhdistäminen

3

matti.tedre@cs.joensuu.fi



Puute pizzaohjelmassa [taas!]

- Pizzamestari Pekka huomaa että pizzaohjelmassa on vielä puute...
- Jos ohjelmalle vahingossa syöttää hirveän isoja pizzan kokoja, ohjelma laskee pizzan neliöhinnan ilman mutinoita
- Kts. Pekan ohjelma

4

matti.tedre@cs.joensuu.fi



```

pii = 3.1415
halk = input("Anna pizzan halkaisija (cm):")
while halk<=0:
    print "Halkaisijan pitää olla yli nolla"
    halk = input("Anna pizzan halkaisija (cm):")
hinta = input("Anna pizzan hinta (e):")
while hinta<=0:
    print "Hinnan pitää olla yli nolla euroa"
    hinta = input("Anna pizzan hinta (e):")
sade = halk / 2.0
sade = sade / 100
ala = pii * sade * sade
hinta_per_m2 = hinta/ala
print "Pizza maksaa", hinta_per_m2, "e/m2"

```

5

matti.tedre@cs.joensuu.fi



Muutokset (jatk.)

- Lisäksi Pekan pitää muuttaa alkuehtoa seuraavasta toistolauseesta:

```

halk = input("Anna pizzan halkaisija (cm):")
while halk<=0:
    print "Halkaisijan pitää olla yli nolla"
    halk = input("Anna pizzan halkaisija (cm):")

```

- Tällä hetkellä käyttäjältä kysellään pizzan kokoa niin kauan kunnes käyttäjä antaa pizzan halkaisijaksi yli nollan.

7

matti.tedre@cs.joensuu.fi



Muutokset

- Pekka päättää muokata ohjelmaansa niin, että ohjelma ei suostu laskemaan pizzan neliöhintaa yli Pekan uunin kokoisille pizzoille.
- Pekan uunin suu on 1.8 metriä.
- Pekka päättää lisätä ohjelmaan muuttujan, jonka tehtävänä on säilyttää tietoa uunin suuaukon koosta. Muuttujan arvo määrätään ohjelman alussa eikä sitä muuteta (kiintoarvo)

```

uunin_suu = 180

```

6

matti.tedre@cs.joensuu.fi



Alkuehto

- Alkuehto tässä ohjelmanpätkässä sanoo jotakuinkin:
 - “niin kauan kuin annettu luku on pienempi tai yhtäsuuri kuin nolla” [kysy lukua uudelleen].
 - Riittää kun käyttäjä antaa arvon joka on yli nolla

```

halk = input("Anna pizzan halkaisija (cm):")
while halk<=0:
    print "Halkaisijan pitää olla yli nolla"
    halk = input("Anna pizzan halkaisija (cm):")

```

8

matti.tedre@cs.joensuu.fi



Uusi alkuehto

- Uuden alkuehdon täytyy kysyä:
- Niin kauan kuin käyttäjä antaa virheellisen pizzan koon, eli
 - *nolla tai alle*
 - **TAI** *yli 180cm*
- ...niin kysy pizzan kokoa uudelleen

9

matti.tedre@cs.joensuu.fi



or

- `or` – operaattorilla voidaan tehdä “tai”-vertailuja
- Uusi alkuehto:
 - niin kauan kuin käyttäjän antama pizzan koko on *nolla tai alle* TAI *yli 180*, toista sisennettyä osaa

```
halk = input("Anna pizzan halkaisija (cm):")
while halk<=0 or halk>180:
    print "Halkaisijan pitää olla yli nolla",
    print "mutta enintään 180 cm"
    halk = input("Anna pizzan halkaisija (cm):")
```

10

matti.tedre@cs.joensuu.fi



Valmis ohjelma

- Korvataan vielä 180 kiintoarvolla `uunin_suu`

```
halk = input("Anna pizzan halkaisija (cm):")
while halk<=0 or halk>uunin_suu:
    print "Halkaisijan pitää olla yli nolla",
    print "mutta enintään", uunin_suu, "cm"
    halk = input("Anna pizzan halkaisija (cm):")
```

- Alkuehto ja sisennetty osuus:
 - “Niin kauan kuin käyttäjän antama halkaisija on joko (1) nolla tai alle, tai (2) yli uunin suuaukon koon,
 - tulosta käyttäjälle ohjeita ja kysy halkaisijaa uudelleen”

11

matti.tedre@cs.joensuu.fi



```
pii = 3.1415
uunin_suu = 180
halk = input("Anna pizzan halkaisija (cm):")
while halk<=0 or halk>uunin_suu:
    print "Halkaisijan pitää olla yli nolla",
    print "mutta enintään ", uunin_suu
    halk = input("Anna pizzan halkaisija (cm):")
hinta = input("Anna pizzan hinta (e):")
while hinta<=0:
    print "Hinnan pitää olla yli nolla euroa"
    hinta = input("Anna pizzan hinta (e):")
sade = halk / 2.0
sade = sade / 100
ala = pii * sade * sade
hintaper_m2 = hinta/ala
print "Pizza maksaa", hintaper_m2, "e/m2"
```

12

matti.tedre@cs.joensuu.fi



Sulkapalloilija Simon ohjelma

- Sulkapalloilija Simolla on huono muisti. Simo ei koskaan muista paljonko pelitilanne on.
- Simo päättää tehdä tietokoneohjelman, joka pitää yllä tietoa pelitilanteesta.
- Peli lähtee nollassa pisteestä ja menee poikki 15 pisteestä (ei tarvitse olla 2 pisteen eroa)

13

matti.tedre@cs.joensuu.fi



Simon Sulkapallo-ohjelma

- Simon ohjelman pitäisi toimia seuraavasti:
 - Pyydetään pelaajan A nimi
 - Pyydetään pelaajan B nimi
 - Molemmat pelaajat aloittavat nollassa
 - Toistetaan seuraavaa kunnes jompi kumpi pelaajista saavuttaa 15 pistettä:
 - Kysy kummalle annetaan piste (a vai b)
 - Lisää yksi piste ko. pelaajalle
 - Tulosta pistetilanne
 - Tulosta voittajan nimi

14

matti.tedre@cs.joensuu.fi



Sulkapallo-ohjelma (1)

- Nimien kysely on helppo tehdä:

```
A_nimi = raw_input("Anna pelaajan A nimi:")  
B_nimi = raw_input("Anna pelaajan B nimi:")
```

- Muuttujien A_nimi ja B_nimi tehtävänä on pitää yllä käyttäjien nimiä.
- Alustuksen jälkeen muuttujien arvot säilyvät samana halki ohjelman suorituksen

15

matti.tedre@cs.joensuu.fi



Sulkapallo-ohjelma (2)

- Luodaan muuttujat nimeltä *A_pisteet* ja *B_pisteet*, joiden tehtävänä on pitää yllä tietoa siitä montako pistettä kullakin pelaajalla on.

```
B_pisteet = 0  
A_pisteet = 0
```

- Aluksi kummallakaan ei ole yhtään pistettä.
- Muuttujat askeltavat nollassa ylöspäin yhden pisteen askelin.

16

matti.tedre@cs.joensuu.fi



Sulkapallo-ohjelma (3)

- Täytyisi luoda toistorakenne, jossa kysellään kumpi sai pisteen kunnes jompi kumpi on päässyt 15 pisteeseen.
- Eli,
 - "Toista sisennettyä osaa (kyselyä) niin kauan kuin pelaajalla A on alle 15 pistettä **JA** pelajalla B on alle 15 pistettä"

- Toisin sanoen

```
while B_pisteet<15 and A_pisteet<15
    ...
```

17

matti.tedre@cs.joensuu.fi



Sulkapallo-ohjelma (4)

- Mitä toistossa tapahtuu?
- Sisennettyä osaa toistetaan niin kauan kun molemmilla on alle 15 pistettä
- Sisennetyssä osassa kysytään kumpi pelaajista (a/b) sai pisteen
- Mikäli a sai pisteen, lisätään a:n laskuria
 - A_pisteet = A_pisteet + 1
- Mikäli b sai pisteen, lisätään b:n laskuria
 - B_pisteet = B_pisteet + 1

18

matti.tedre@cs.joensuu.fi



Sulkapallo-ohjelma (5)

- Miten tehdään kysely ja vertailu?
- Luetaan ensiksi käyttäjältä tieto siitä kumpi sai pisteen:

```
syote = raw_input("Kumpi sai pisteen (a/b/A/B)?")
```

19

matti.tedre@cs.joensuu.fi



Sulkapallo-ohjelma (6)

- Sitten tarkastellaan käyttäjän antamaa merkkijonoa.
- Jos käyttäjän antoi merkin "a" **TAI** merkin "A"
 - niin kasvatetaan pelaajan A pisteitä

```
if syote=="a" or syote=="A":
    A_pisteet = A_pisteet + 1
```

20

matti.tedre@cs.joensuu.fi



Sulkapallo-ohjelma (7)

- Tarkastellaan uudelleen käyttäjän antamaa merkkijonoa.
- Jos käyttäjä antoi merkin "b" **TAI** merkin "B"
 - niin kasvatetaan pelaajan B pisteitä

```
if syote=="b" or syote=="B":  
    B_pisteet = B_pisteet + 1
```

21

matti.tedre@cs.joensuu.fi



```
voittopist=15  
A_nimi = raw_input("Anna pelaajan A nimi:")  
B_nimi = raw_input("Anna pelaajan B nimi:")  
B_pisteet = 0  
A_pisteet = 0  
  
while B_pisteet<voittopist and A_pisteet<voittopist:  
    syote = raw_input("Kumpi sai pisteen (a/b/A/B)?")  
    if syote=="a" or syote=="A":  
        A_pisteet = A_pisteet + 1  
    if syote=="b" or syote=="B":  
        B_pisteet = B_pisteet + 1  
    print "Pistetilanne:"  
    print A_nimi, A_pisteet, "/",  
    print B_nimi, B_pisteet  
  
if B_pisteet>A_pisteet:  
    print B_nimi, "voitti!"  
elif A_pisteet>B_pisteet:  
    print A_nimi, "voitti!"
```

23

matti.tedre@cs.joensuu.fi



Sulkapallo-ohjelma (8)

- Kunkin toistokerran lopuksi tulostetaan kulloinenkin pistetilanne:

```
print "Pistetilanne:"  
print A_nimi, A_pisteet, "/",  
print B_nimi, B_pisteet
```

- Esim:

```
Pistetilanne:  
Matti 5 / Pekka 12
```

22

matti.tedre@cs.joensuu.fi



Sisäkkäinen rakenne

- Sulkapallo-ohjelmassa on kaksi rakennetta sisäkkäin:
 - Alkuehtoinen toistolause (while)
 - Toistolauseeseen sisällä ehtolause (if)

```
while B_pisteet<voittopist and A_pisteet<voittopist:  
    syote = raw_input("Kumpi sai pisteen (a/b/A/B)?")  
    if syote=="a" or syote=="A":  
        A_pisteet = A_pisteet + 1  
    if syote=="b" or syote=="B":  
        B_pisteet = B_pisteet + 1  
    print "Pistetilanne:"  
    print A_nimi, A_pisteet, "/",  
    print B_nimi, B_pisteet
```

24

matti.tedre@cs.joensuu.fi



Sisäkkäisiä rakenteita

25

matti.tedre@cs.joensuu.fi



Sepon Salamaohjelma

- Sepolla on ohjelma joka pyytää käyttäjää syöttämään tiedon siitä miten kauan kesti salaman välähdyksestä äänen jyrähdykseen
- Sepon ohjelma kysyy käyttäjältä aikoja kunnes käyttäjä antaa negatiivisen arvon.
- Seppo haluaisi lisätä ohjelmaansa muutamia ominaisuuksia

26

matti.tedre@cs.joensuu.fi



Uudet ominaisuudet

- Sepon uuden ohjelman pitäisi ohjelman päätteeksi kertoa
 - *montako salamaa* käyttäjä syötti
 - kuinka lähelle *lähin* salama iski
- Seppo tarvitsee ohjelmaansa kaksi muuttujaa joilla on erityiset tehtävät:
 - Toisen muuttujan täytyy pitää yllä tietoa siitä montako salamaa on syötetty
 - Toisen muuttujan täytyy pitää yllä tietoa tähän mennessä lähimmäksi iskeneestä salamasta

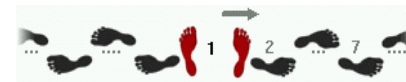
27

matti.tedre@cs.joensuu.fi



Muuttujat

- Seppo lisää muuttujan *salamat_lkm*, jonka tehtävänä on pitää yllä tietoa siitä montako salamaa käyttäjä on syöttänyt
 - Lähtee liikkeelle nolasta
 - Joka kerta kun käyttäjä syöttää *hyväksyty*n ajan, kasvatetaan yhdellä



- Muuttujan rooli on *askeltaja* (askeltaa kokonaislukuja nolasta ylöspäin)

28

matti.tedre@cs.joensuu.fi



Muuttujat (jatk.)

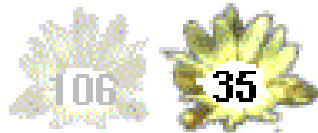
- Seppo lisää myös muuttujan `salamat_lahin`, jonka tehtävänä on pitää yllä tietoa siitä kuinka kaukana *lähimmäksi osunut* salama löi.
 - Aina kun käyttäjä on syöttänyt järkevän ajan, lasketaan etäisyys (*etaisyys*), ja tehdään vertailu:
 - “Oliko juuri lyönyt salama lähempänä kuin tähän mennessä lähin?”
 - (“Onko muuttujan *etaisyys* arvo vähemmän kuin muuttujan *salamat_lahin* arvo?”)
 - Mikäli juuri lyönyt salama oli lähempänä kuin muuttujaan *salamat_lahin* talletettu etäisyys, niin päivitetään muuttujan *salamat_lahin* arvoa.

29

matti.tedre@cs.joensuu.fi



Muuttujat (jatk.)



- Muuttujan *salamat_lahin* rooli tässä ohjelmassa on *sopivimman säilyttäjä*
 - “Sopivin” tässä tapauksessa on pienin arvo, eli lähimmäksi osuneen salaman etäisyys Seposta
 - Millainen arvo muuttujalle *salamat_lahin* pitäisi antaa ohjelman suorituksen alussa, ennen kuin yhtään salamaa ollaan luettu?

31

matti.tedre@cs.joensuu.fi



Ohjelmakoodi

- Seuraava ohjelmakoodi toiminee:
 - Jos kyseessä on ensimmäinen salama, se on luonnollisestikin lähin
 - Jos juuri syötetyn salaman etäisyys oli lähempänä kuin muistissa oleva lähin salama, niin laitetaan muistiin tämä juuri syötetty salama.

```
if salamat_lkm==1:
    salamat_lahin=etaisyys
if etaisyys<salamat_lahin:
    salamat_lahin=etaisyys
```

30

matti.tedre@cs.joensuu.fi



```
aanen_nopeus = 340.0 # kiintoarvo
salamat_lkm = 0 # askeltaja
salamat_lahin = -1 # sopivimman säilyttäjä
```

```
print "Syötä alle 0 lopettaaksesi"
aika = input("Anna kulunut aika (s):")
while aika>=0:
    etaisyys = aika * aanen_nopeus
    etaisyys = etaisyys / 1000
    salamat_lkm = salamat_lkm+1
    if salamat_lkm==1:
        salamat_lahin=etaisyys
    if etaisyys<salamat_lahin:
        salamat_lahin=etaisyys
    print "Salama löi ", etaisyys, "km:n päässä"
    aika = input("Anna kulunut aika (s):")
print "Salamoita löi", salamat_lkm, "kappaletta"
if salamat_lkm>0:
    print "Lähin osui", salamat_lahin, "km:n päässä"
```

32

matti.tedre@cs.joensuu.fi



Sisäkkäiset rakenteet

- Tässä Sepon uudessa ohjelmassa on erilaisia ohjelmointikielen rakenteita sisäkkäin
 - Alkuehtoinen toisto (while)
 - 2 ehtolausetta (if)
- Kuten ollaan jo opittu, yksi rakenne (esim. toistettava osa tai ehdollisesti suoritettava osa) erotetaan sisennyksellä
- Rakenteen sisällä, sisennytyssä osassa, voi olla toinen rakenne, joka erotaan vielä syvemmällä sisennyksellä

33

matti.tedre@cs.joensuu.fi



34

```
aanen_nopeus = 340.0 # kiintoarvo
salamat_lkm = 0 # askeltaja
salamat_lahin = 0 # sopivimman säilyttäjä

print "Syötä alle 0 lopettaaksesi"
aika = input("Anna kulunut aika (s):")
while aika>=0:
    etaisyyys = aika * aanen_nopeus
    etaisyyys = etaisyyys / 1000
    salamat_lkm = salamat_lkm+1
    if salamat_lkm==1:
        salamat_lahin=etaisyyys
    elif etaisyyys<salamat_lahin:
        salamat_lahin=etaisyyys
    print "Salama löi ", etaisyyys, "km:n päässä"
    aika = input("Anna kulunut aika (s):")
print "Salamoita löi", salamat_lkm, "kappaletta"
if salamat_lkm>0:
    print "Lähin osui", salamat_lahin, "km:n pää
```

matti.tedre@cs.joensuu.fi

Kaksi ehtolausetta?

```
if salamat_lkm==1:
    salamat_lahin=etaisyyys
elif etaisyyys<salamat_lahin:
    salamat_lahin=etaisyyys
```

- Voitaisiinko sama kirjoittaa käyttäen vain yhtä ehtolausetta?

35

matti.tedre@cs.joensuu.fi

Lisää loogisista operaattoreista

36

matti.tedre@cs.joensuu.fi

and ja or

- **and**-operaattorilla voidaan niputtaa vertailuja yhteen niin että niiden *kaikkien* täytyy toteutua jotta vertailulauseke on tosi:

- Kuukauden täytyy olla vähintään yksi ja enintään kaksitoista ($1 \leq kk \leq 12$):

```
if kk>=1 and kk<=12:  
    print "Syöte kelpaa"
```

- Saadakseen opintotukea, ammatin täytyy olla opiskelija ja ikää pitää olla 18 vuotta:

```
if ammatti=="opiskelija" and ika>=18:  
    print "beep. opintotuki myönnetty"
```

37

matti.tedre@cs.joensuu.fi



and ja or

- Vaihto-opiskelua varten täytyy olla vähintään 2. vuoden opiskelija ja opintopisteitä vähintään 40:

```
if opintovuosi>=2 and op>=40:  
    print "Tilataanko jo liput?"
```

- Jotta käyttäjä voi liittyä Mönin Riennon D-juniorijoukkueeseen, täytyy useita ehtoja täytyä:

```
if kotipaikka=="Mönni" and ika<15 and ika>=12:  
    print "Tervetuloa mukaan!"  
else:  
    print "Sori...."
```

38

matti.tedre@cs.joensuu.fi



and ja or

- **or**-operaattorilla voidaan niputtaa vertailuja yhteen niin että *jonkin* vertailuista täytyy toteutua jotta vertailulauseke on tosi:

- Jos käyttäjä syötti kuukaudeksi alle 1 tai yli 12, kysy kuukautta uudelleen:

```
if kk<1 or kk>12:  
    print "Kuukauden pitää olla välillä 1..12!"
```

- Jotta baariin pääsee jonon ohi, täytyy olla joko kanta-asiakas tai varaa maksaa kunnan tipit:

```
if kanta_asiakas==True or rahaa>50:  
    print "Hei siellä, teitä odotetaan sisällä!"
```

39

matti.tedre@cs.joensuu.fi



and ja or

- Vuoristorataan pääsee jos ikää on vähintään 15 vuotta tai jos pituutta on yli 140 cm:

```
if ika>=15 or pituus>140:  
    print "Viuh!"
```

- Jotta pääsisi Ivy League-yliopistoon, täytyy joko olla hyvä todistus (GPA yli 4) tai oikeat sukujuuret:

```
if GPA>4.0 or isin_ammatti=="Presidentti":  
    print "Tervetuloa Yaleen!"  
else:  
    print "Mene muualle"
```

40

matti.tedre@cs.joensuu.fi



and ja or

- Sulkumerkeillä voidaan tehdä vielä monipuolisempia vertailuja:
- Esimerkiksi: Tämän kurssin pääsee läpi jos jompi kumpi seuraavista toteutuu:
 - Yleisessä tentissä saa vähintään 15/30 pistettä **TAI**
 - Vähintään 10 demotehtävää on tehty **JA** loppukokeesta saa vähintään 14/30 pistettä

```
if yl_tentti>=15 or (demopts>=10 and loppuk>=14):  
    print "3 op suoritettu."
```

41

matti.tedre@cs.joensuu.fi



and ja or

- Sama asia voidaan kirjoittaa eri tavoilla!
- Esimerkiksi: käsitellään syötettä jossa käyttäjän täytyy antaa luku välillä 1..356:

42

matti.tedre@cs.joensuu.fi



Syöte jonka pitää olla välillä 1..356

- Voidaan kirjoittaa

```
paiva = input("Syötä päivän numero: ")  
if paiva<1 or paiva>356:  
    print "Virhe! Anna luku väliltä 1..356"  
else:  
    print "Ok! Syöte oli välillä 1..356"
```

- Erilainen versio:

```
paiva = input("Syötä päivän numero: ")  
if paiva>=1 and paiva<=356:  
    print "Ok! Syöte oli välillä 1..356"  
else:  
    print "Virhe! Anna luku väliltä 1..356"
```

43

matti.tedre@cs.joensuu.fi



Syöte jonka pitää olla välillä 1..356

- Vielä erilainen versio:

```
paiva = input("Syötä päivän numero: ")  
if paiva>0 and paiva<357:  
    print "Ok! Syöte oli välillä 1..356"  
else:  
    print "Virhe! Anna luku väliltä 1..356"
```

44

matti.tedre@cs.joensuu.fi



Lisää sisäkkäisiä rakenteita

45

matti.tedre@cs.joensuu.fi



Ritvan ohjelma

- Rehtori Ritvalla on ohjelma jolla voidaan laskea kurssien keskiarvoja
- Ohjelma pyytää ensiksi käyttäjältä arvosanojen määrän *kurssit_lkm*
- Sitten ohjelma pyytää käyttäjältä niin monta arvosanaa kuin kursseja on
- Lopuksi ohjelma laskee ja tulostaa kurssien keskiarvon

46

matti.tedre@cs.joensuu.fi



Ritvan ohjelma, versio 2

```
# Useamman kurssin keskiarvon laskeva ohjelma
# summa: kokooja
# kurssi: askeltaja
# keskiarvo, kehote: tilapäissäilö
# arvosana: tuoreimman säilyttäjä
# kurssit_lkm: tuoreimman säilyttäjä

summa = 0
kurssit_lkm = input("Montako arvosanaa?")
for kurssi in range(1, kurssit_lkm+1):
    print "Kurssi", kurssi
    arvosana = input("Anna arvosana: ")
    summa = summa + arvosana
keskiarvo = float(summa) / kurssit_lkm
print "Keskiarvo on", keskiarvo
```

47

matti.tedre@cs.joensuu.fi



Puutteita Ritvan ohjelmassa

- Ritvan ohjelmassa on kuitenkin puutteita
- Käyttäjä voi syöttää ohjelmalle kurssiarvosanoja jotka eivät ole sinne päinkään oikein
- Ohjelma laskee keskiarvon vaikka kurssiarvosanat olisivat virheellisiäkin
- Pitäisi muokata ohjelmaa niin että arvosanoiksi hyväksytään pelkästään lukuja välillä 4-10.

48

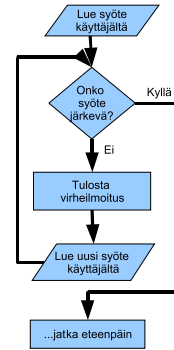
matti.tedre@cs.joensuu.fi



Mitä on opittu edellisistä ohjelmista?

- Pizzaohjelmasta:

- Syötteen tarkastelu ja syötteen lukeminen aina uudestaan kunnes syöte on järkevä



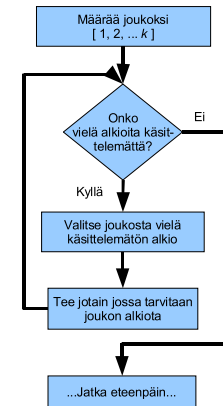
49

matti.tedre@cs.joensuu.fi

Mitä on opittu edellisistä ohjelmista?

- Ritvan edellisestä ohjelmasta:

- Toistorakenne, jossa muuttuja askeltaa kaikki joukon jäsenet läpi yksi kerrallaan

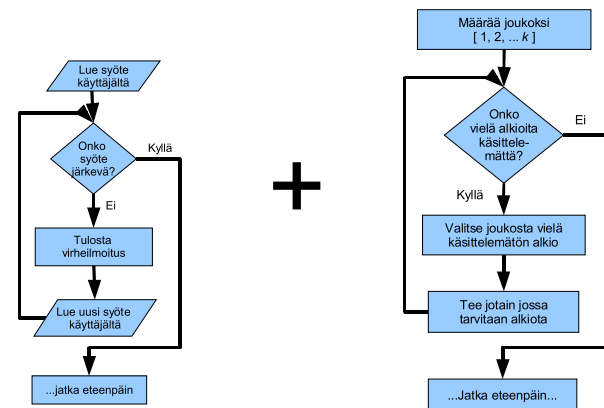


50

matti.tedre@cs.joensuu.fi

Yhdistetään jo opitut rakenteet!

- Tehdään ohjelma joka käy läpi kaikki kurssit...
 - Rakenne Ritvan ohjelmasta
- ...ja joka jokaisen kurssin kohdalta varmistaa että käyttäjä on syöttänyt oikeanlaista tietoa
 - Rakenne pizzaohjelmasta

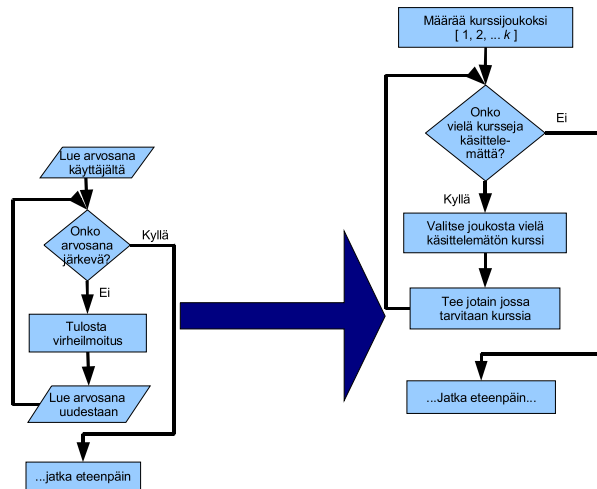


51

matti.tedre@cs.joensuu.fi

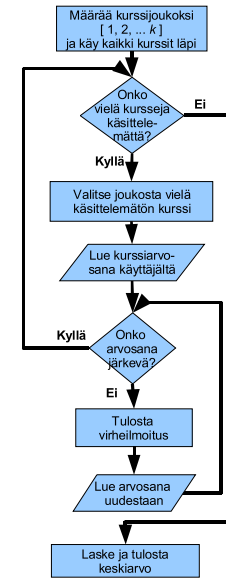
52

matti.tedre@cs.joensuu.fi



53

matti.tedre@cs.joensuu.fi



54

matti.tedre@cs.joensuu.fi

- Keskiarvon laskeminen

```

summa = 0

kurssit_lkm = input ("Montako arvosanaa?")
for kurssi in range(1,kurssit_lkm+1):
    print "Käsitellään kurssia", kurssi
    arvosana = input ("Anna arvosana: ")
    summa = summa + arvosana
keskiarvo = float(summa) / kurssit_lkm
print "Keskiarvo on", keskiarvo
  
```

- Oikeellisen kurssiarvosanan pyytäminen

```

alin=4
ylin=10

arvosana = input ("Anna arvosana: ")
while arvosana<alin or arvosana>ylin:
    print "Arvosanan pitää olla ", alin, "-", ylin
    arvosana = input ("Anna arvosana: ")
  
```

55

matti.tedre@cs.joensuu.fi

Sisäkkäiset rakenteet

- Ritvan ohjelmassa erotetaan askeltavan toistorakenteen toistettava osa sisennyksellä
- Askeltava (for) toisto käy kaikki kurssit läpi
- Ritvan ohjelmaan halutaan myös toistorakenne jossa kurssiarvosanaa kysellään niin kauan kunnes käyttäjä antaa 4-10.
- Erotetaan tuo toistorakenne vielä syvemmillä sisennyksellä!

56

matti.tedre@cs.joensuu.fi

Kaksi sisäkkäistä toistoa:

```
summa = 0
alin=4
ylin=10
kurssit_lkm = input ("Montako arvosanaa?")

for kurssi in range(1,kurssit_lkm+1):
    print "Käsitellään kurssia", kurssi
    arvosana = input ("Anna arvosana: ")
    while arvosana<alin or arvosana>ylin:
        print "Arvosanan pitää olla ", alin, "-", ylin
        arvosana = input ("Anna arvosana: ")
    summa = summa + arvosana
keskiarvo = float(summa) / kurssit_lkm
print "Keskiarvo on", keskiarvo
```

57

matti.tedre@cs.joensuu.fi



Valmis ohjelma:

```
summa = 0
alin=4
ylin=10
kurssit_lkm = input ("Montako arvosanaa?")

for kurssi in range(1,kurssit_lkm+1):
    print "Käsitellään kurssia", kurssi
    arvosana = input ("Anna arvosana: ")
    while arvosana<alin or arvosana>ylin:
        print "Arvosanan pitää olla ", alin, "-", ylin
        arvosana = input ("Anna arvosana: ")
    summa = summa + arvosana
keskiarvo = float(summa) / kurssit_lkm
print "Keskiarvo on", keskiarvo
```

58

matti.tedre@cs.joensuu.fi



Sepon salamaohjelma [taas]

- Sepolla on kaksi ohjelmaa:
 - 1)ohjelma joka osaa laskea ajan perusteella etäisyyksiä kunnes käyttäjä antaa negatiivisen luvun
 - 2)ohjelma joka osaa analysoida osuiko salama
 - Kohdalle
 - Lähelle
 - Kauas

59

matti.tedre@cs.joensuu.fi



Sepon salamaohjelma versio 5

- Seppo haluaa muuttaa ohjelmaansa niin, että ohjelma
 - Laskee jokaisen salaman etäisyyden
 - Analysoi jokaisen salaman (osuiko kohdalle, lähelle vai kauas)
 - Kyselee salamoita kunnes käyttäjä antaa *negatiivisen* luvun
- Täytyy yhdistää Sepon kaksi ohjelmaa

60

matti.tedre@cs.joensuu.fi



Sepon salama-analyysi

```
aanen_nopeus = 340.0
lahella = 1.5
```

```
aika = input("Anna kulunut aika (s):")
etaisyys = aika * aanen_nopeus
etaisyys = etaisyys / 1000
```

```
if etaisyys<0:
    print "Syötä positiivinen luku"
elif etaisyys==0:
    print "Salama osui kohdalle."
elif etaisyys<=lahella:
    print "Salama osui lähelle."
else:
    print "Salama osui kauas."
print "(C) Seppo 2006"
```

61

matti.tedre@cs.joensuu.fi

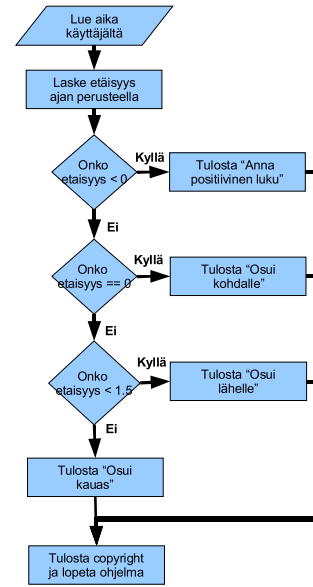
Sepon ohjelma, versio 3

```
aanen_nopeus = 340.0
print "Syötä 0 tai alle lopettaaksesi"
```

```
aika = input("Anna kulunut aika (s):")
while aika>0:
    etaisyys = aika * aanen_nopeus
    etaisyys = etaisyys / 1000
    print "Salama löi ", etaisyys, "km:n päässä"
    aika = input("Anna kulunut aika (s):")
print "(C) Seppo 2006"
```

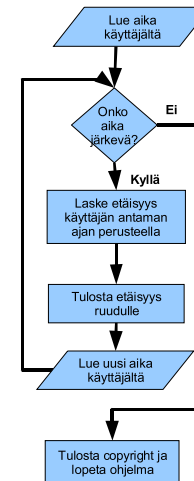
63

matti.tedre@cs.joensuu.fi



62

matti.tedre@cs.joensuu.fi



64

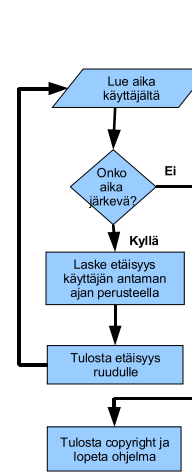
matti.tedre@cs.joensuu.fi

Yhdistetään Sepon ohjelmat

- Otetaan Sepon kaksi ohjelmaa, ja yhdistetään niistä sellainen ohjelma joka
 - Toistaa kyselyä kunnes käyttäjä syöttää negatiivisen luvun (alkuehtoinen toisto)
 - Analysoi salaman etäisyyden (ehtolause)

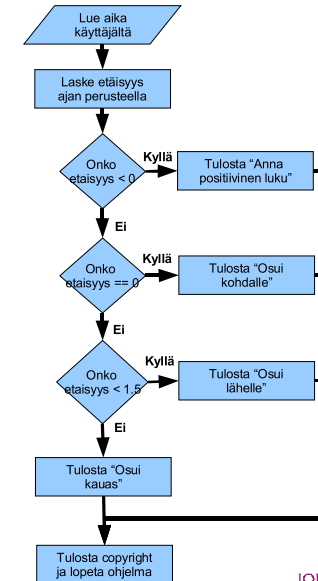
65

matti.tedre@cs.joensuu.fi



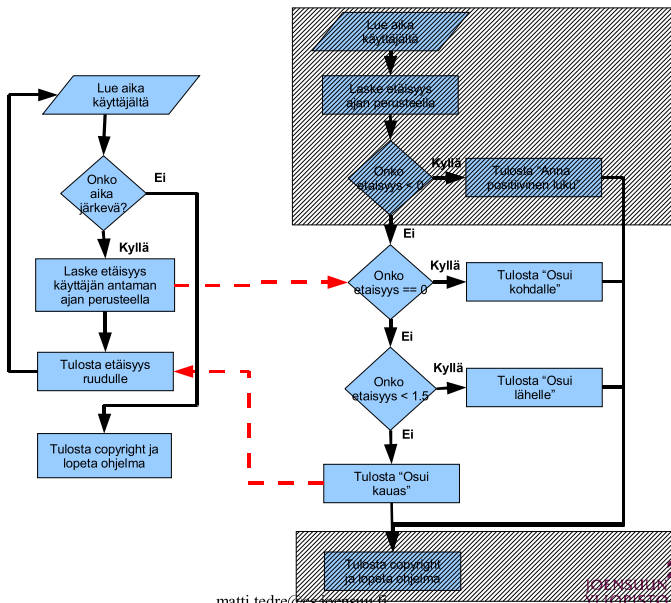
66

matti.tedre@cs.joensuu.fi



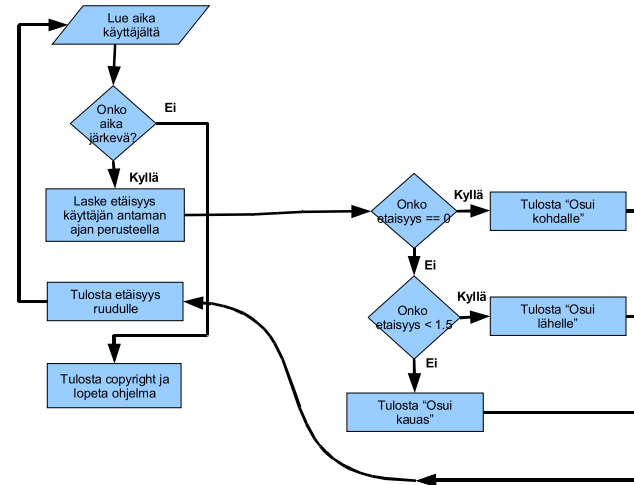
67

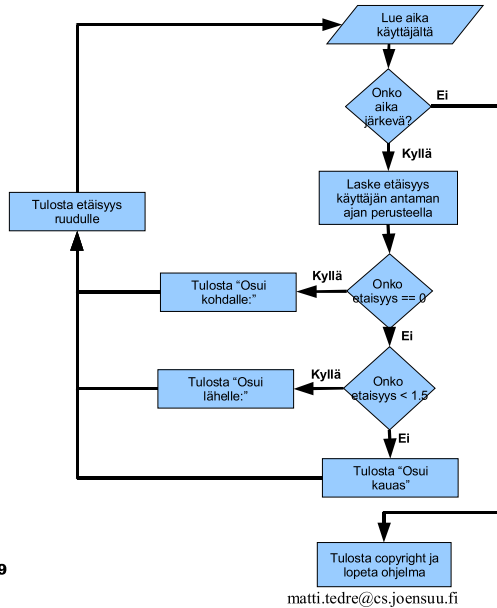
matti.tedre@cs.joensuu.fi



68

matti.tedre@cs.joensuu.fi





```

aanen_nopeus = 340.0 # m/s
kaukana = 1.5 # km

print "Salama-analyysiohjelma"

aika = input("Anna kulunut aika (s):")
while aika>=0:
    etaisyyys = aika * aanen_nopeus
    etaisyyys = etaisyyys / 1000
    if etaisyyys==0:
        print "Salama löi kohdalle:",
    elif etaisyyys<kaukana:
        print "Salama löi lähelle:",
    else:
        print "Salama löi kauas:",
        print etaisyyys, "km:n päähän"
    aika = input("Anna kulunut aika (s):")
print "(C) Seppo 2006"

```

```

aanen_nopeus = 340.0 # m/s
kaukana = 1.5 # km

```

```

print "Salama-analyysiohjelma"
print "Anna kulunut aika (0 tai alle lopettaa)"

```

```

aika = input("Anna kulunut aika (s):")
while aika>=0:
    etaisyyys = aika * aanen_nopeus
    etaisyyys = etaisyyys / 1000
    if etaisyyys==0:
        print "Salama löi kohdalle:",
    elif etaisyyys<kaukana:
        print "Salama löi lähelle:",
    else:
        print "Salama löi kauas:",
    print etaisyyys, "km:n päähän"
    aika = input("Anna kulunut aika (s):")
print "(C) Seppo 2006"

```

Tyypillisiä virheitä

Päättymätön toisto

- Silloin tällöin huonosti suunnitellut ohjelmat jäävät *ikuisen silmukkaan*. Esim. ohjelma joka tulostaa ruudulle n kappaletta asteriskeja *:

```
nro = input("Kuinka monta asteriskia tulostetaan: ")
laskuri=0
while laskuri!=nro:
    laskuri = laskuri + 1
    print "*"

```

- Millaisilla syötteillä ohjelma jää "ikuisen silmukkaan"?

73 – Miten korjataan?

matti.tedre@cs.joensuu.fi



Vailinainen ehtorakenne

- Yleensä ehtorakenteissa pitäisi olla haara jokaiselle mahdolliselle vaihtoehdolle.

```
opiskelijoita = input("Montako ilmoittautui: ")
if opiskelijoita<80:
    print "Varataan sali M2"
elif opiskelijoita>80:
    print "Varataan sali M1"

```

- Milloin ohjelma toimii oudosti?
 - Miten korjataan?

74

matti.tedre@cs.joensuu.fi



"Yhdellä hutii"

- Usein ohjelmat toimivat "melkein" oikein. Esim. ohjelma joka tulostaa halutun numeron kertotaulun 1-10:

```
nro = input("Anna luku niin laskea sen kertotaulun: ")
laskuri = 1
while laskuri<10:
    print laskuri, "*", nro, "=", laskuri*nro
    laskuri = laskuri + 1

```

- Miksi ohjelma on väärin?
 - Miten korjataan?

75

matti.tedre@cs.joensuu.fi



Esimerkki: kertotaulun tulostus

- Luennoilla mietittiin että miten tulostusta voidaan muotoilla paremmaksi. Tässä esimerkki neljän levyisistä sarakkeista (*ei kuulu varsinaisesti kurssille!*):

```
rivi = 1
while rivi<=10:
    sarake = 1
    while sarake<=10:
        print "%4.d" % (rivi*sarake),
        sarake = sarake+1
    rivi = rivi + 1
print

```

76

matti.tedre@cs.joensuu.fi

