

# Johdatus ohjelmointiin

Syksy 2006

Viikko 3

20.9. - 21.9.

1

matti.tedre@cs.joensuu.fi



2

matti.tedre@cs.joensuu.fi



## Tällä viikolla käsiteltävät asiat

- Tiedon käsittely ja yksinkertaiset päättelyt
  - Peruslaskutoimitukset
  - Vertailu
  - Ehtolauseet
  - Ohjelman haarautuminen

## Harjoittelu

- Tärkeintä on se että *kokeilee* erilaisia yksinkertaisia ohjelmia kotona!
- Näitä asioita on vaikeaa oppia pelkästään lukemalla mutta harjoitellen oppiminen on helppoa
- Kurssin kotisivuilta on linkki verkosta saatavaan Pythonin Windows-versioon

3

matti.tedre@cs.joensuu.fi



4

matti.tedre@cs.joensuu.fi



## Tiedon käsittely

[Kuva]

## Esimerkkiohjelma

- Säättökija Seppo tutkii ukkosta ja Seppo joutuu usein laskemaan kuinka kaukana salamat iskevät
- Seppo laskee etäisyyden ukkosrintamaan mittaamalla ajan salaman välähdyksestä jyrinän kuulumiseen
- Seppo haluaisi automatisoida nämä laskutoimitukset tietokoneella

5

matti.tedre@cs.joensuu.fi



## Ohjelman kuvaus

- Ohjelma saa syötteenään ajan välähdyksestä jyrinään (sekunteina) ja laskee kuinka kaukana salama iski
- Ohjelman pitää tulostaa etäisyys kilometreinä

6

matti.tedre@cs.joensuu.fi



## Algoritmi

1. Pyydä käyttäjältä aika sekunteina
2. Laske kuinka kaukana salama iski
3. Tulosta vastaus käyttäjälle
4. Tulosta ilmoitus että ohjelman suoritus loppui

7

matti.tedre@cs.joensuu.fi



## Kiintoarvot

- Tiedetään seuraavat asiat:
  - Äänen nopeus (20°C) on 340 m/s
- Määritellään kiintoarvoksi  
`aaenen_nopeus = 340.0`
- ...Miksi desimaalipiste?
- ...Miksi kiintoarvo?
  - Jos Seppo haluaa laskea etäisyyksiä myös talvella, täytyy kiintoarvoa muuttaa (-10°C:ssä äänen nopeus on enää 325 m/s)

8

matti.tedre@cs.joensuu.fi



## Tiedon lukeminen

- Pyydä käyttäjältä aika sekunteina ja sadasosina

```
aika = input("Anna kulunut aika:")
```

- Talletetaan tieto muuttujaan nimeltä `aika`.
- Muuttujan `aika` tehtävänä on säilyttää käyttäjältä luettua tietoa
- Mitä tapahtuu jos käyttäjä antaa numeron sijaan merkkijonon?

9

matti.tedre@cs.joensuu.fi



## Tiedon käsittely

- Lasketaan etäisyys kertomalla aika äänen nopeudella

```
etaisyys = aika * aanen_nopeus
```

- Jolloin saadaan etäisyys metreinä
- Talletetaan tulos muuttujaan `etaisyys`.
- Muuttujan `etaisyys` tehtävänä on säilyttää tietoa väliaikaisesti
- Etäisyys täytyy vielä muuttaa kilometreiksi

```
etaisyys = etaisyys / 1000
```

10

matti.tedre@cs.joensuu.fi



## Tiedon tulostus

- Tulostetaan käyttäjän tarvitsema tieto:

```
print "Salama löi", etaisyys, "kilometrin  
päässä"
```

- Tulostetaan lopuksi ilmoitus siitä että ohjelman suoritus loppui:

```
print "(C) Seppo 2006"
```

11

matti.tedre@cs.joensuu.fi



## Ohjelmalistaus

```
# äänen nopeus: kiintoarvo  
# aika: tuoreimman säilyttäjä  
# etäisyys: tilapäissäilö  
aanen_nopeus = 340.0  
aika = input("Anna kulunut aika:")  
etaisyys = aika * aanen_nopeus  
etaisyys = etaisyys / 1000  
print "Salama löi ", etaisyys, "km:n päässä"  
print "(C) Seppo 2006"
```

12

matti.tedre@cs.joensuu.fi



## Muuttujien roolit

- Muttujan `aanen_nopeus` rooli tässä ohjelmassa on *kiintoarvo*



- Muuttujan `aanen_nopeus` arvoa ei muuteta sen alustamisen jälkeen.
- Sille annetaan siis arvo heti ohjelman alussa eikä arvoa enää muuteta

13

matti.tedre@cs.joensuu.fi



## Muuttujien roolit

- Muttujan `aika` rooli tässä ohjelmassa on *tuoreimman säilyttäjä*



- Muuttujaa `aika` käytetään säilyttämään käyttäjän syöttämää tietoa
- (Tässä ohjelmassa tieto tosin luetaan vain kerran)

14

matti.tedre@cs.joensuu.fi



## Muuttujien roolit

- Muttujan `etaisyys` rooli tässä ohjelmassa on *tilapäissäilö*



- Muuttujaa `etaisyys` käytetään ohjelman selventämiseen, mutta ohjelma voitaisiin myös tehdä ilman tilapäissäilöä

15

matti.tedre@cs.joensuu.fi



## Ohjelma ilman tilapäissäilöä *etaisyys*

```
# äänen nopeus: kiintoarvo
# aika: tuoreimman säilyttäjä
aanen_nopeus = 340.0
aika = input("Anna kulunut aika:")
print "Salama löi ",
print (aika * aanen_nopeus) / 1000,
print "km:n päässä"
print "(C) Seppo 2006"
```

16

matti.tedre@cs.joensuu.fi



# Ohjelman rakenne

## Yksinkertaisen ohjelman rakenne

- Hyvä nyrkkisääntö yksinkertaisten ohjelmien rakentamiseen:
- Ohjelmassa pidetään erillään
  - Kiintoarvojen alustus
  - Tiedon lukeminen
  - Tiedon käsitteleminen
  - Tiedon tulostus

17

matti.tedre@cs.joensuu.fi



## Ohjelmalistaus

```
10 aanen_nopeus = 340.0
35 aika = input("Anna kulunut aika:")
78 etaisyys = aika * aanen_nopeus
   etaisyys = etaisyys / 1000
   print "Salama löi ", etaisyys, "km:n päässä"
   print "(C) Seppo 2006"
```

19

matti.tedre@cs.joensuu.fi

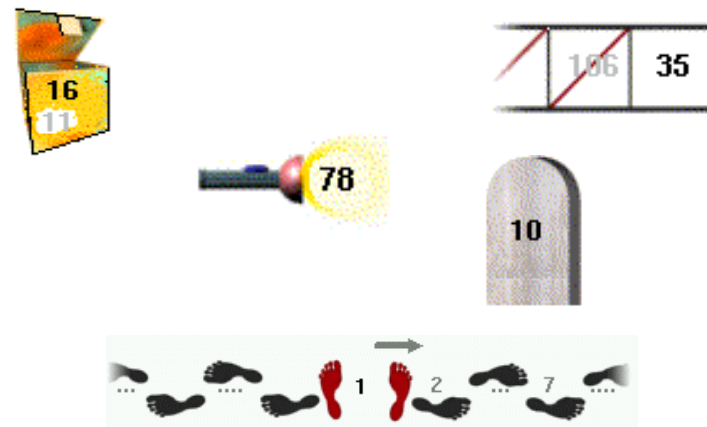


18

matti.tedre@cs.joensuu.fi



## Muuttujien tyypit



20

matti.tedre@cs.joensuu.fi



## Kokonais- ja desimaaliluvut

- Sepon ohjelmassa on määritelty että äänen nopeus on 340.0 m/s
- Desimaalipiste on tärkeä – desimaalipiste määrää että muuttuja on desimaaliluku (eli sillä on kokonaisosa ja desimaaliosa)
- Tämä on tärkeää esimerkiksi jakolaskuissa:
  - Kokonaislukujen jakolaskussa tulos on kokonaisluku + jakojäännös
  - Desimaalilukujen jakolaskussa tulos on desimaaliluku

21

matti.tedre@cs.joensuu.fi



## Jakolasku kokonaisluvuilla

- Sepon ohjelmassa muunnetaan metrit kilometreiksi jakamalla tulos tuhannella.
- Jos käyttäjä on antanut ajaksi 1 sekunti, etäisyys on 340 metriä
- Jaetaan kokonaisluku 340 kokonaisluvulla 1000:

```
>>> print 340/1000
0
>>>
```
- Vastaus on kokonaisluku 0 (jakojäännös on 340)

22

matti.tedre@cs.joensuu.fi



## Jakolasku desimaaliluvuilla

- Edelleen: jos käyttäjä on antanut ajaksi 1 sekunti, etäisyys on 340 metriä
- Jaetaan desimaaliluku 340.0 kokonaisluvulla 1000:

```
>>> print 340.0/1000
0.34
>>>
```

- Vastaus on desimaaliluku 0.34

23

matti.tedre@cs.joensuu.fi



## Miten muuttujan tyyppi määräytyy?

- Mikäli kaikki laskutoimituksen muuttujat ovat kokonaislukutyyppiä, lopputulos on kokonaisluku
- Mikäli laskutoimituksessa on yhtään desimaalilukua, lopputulos on desimaaliluku

24

matti.tedre@cs.joensuu.fi



## Tyypin määräminen

- Muuttujan tyyppiä voi tarvittaessa muuttaa komennoilla `int()` ja `float()`
- Esimerkiksi:  

```
>>> print float(340)/1000  
0.34  
>>>
```
- Ylläolevassa laskutoimituksessa käytetään kokonaisluvun 340 sijaan liukulukua 340.0
- Tästä lisää myöhemmin

25

matti.tedre@cs.joensuu.fi



## Lisää esimerkkejä

26

matti.tedre@cs.joensuu.fi



## Esimerkkiohjelma

- Pizzayrittäjä Pekka miettii hinnoitteluaan. Pekan pitäisi laskea paljonko erikokoisten pizzojen tulisi maksaa
- Asiakkaan pitäisi saada ison pizzan ostaessaan enemmän syötävää suhteessa hintaan
- Perhepizzan ostajan pitäisi saada enemmän pizzaa per euro kuin normaalikokoisen pizzan ostajan
- Pekka päättää tehdä tietokoneohjelman joka laskee pizzan hinnan per neliösenttimetri

27

matti.tedre@cs.joensuu.fi



## Esitiedot

- Pizzan ala voidaan laskea ympyrän pinta-alan kaavalla, joka löytyy esim. MAOL-taulukoista:

$$\textit{Pizzan ala} = \pi \cdot (\textit{pizzan säde})^2$$

- Tai matemaattisemmin  $A = \pi \cdot r^2$   
– (Säde on puolet halkaisijasta)

28

matti.tedre@cs.joensuu.fi



## Ohjelman rakenne

1. Pyydä käyttäjältä pizzan hinta euroina ja pizzan halkaisija senttimetreinä
2. Laske pizzan hinta per neliometri
3. Tulosta käyttäjälle hinta per neliometri (€/m<sup>2</sup>)

29

matti.tedre@cs.joensuu.fi



## Kiintoarvot

- Ohjelmassa tarvitaan piin ( $\pi$ ) likiarvoa
- Alustetaan kiintoarvo:  

```
# pii: kiintoarvo  
pii = 3.1415
```
- ...Miksi kiintoarvona?
  - Myöhemmin halutaan ehkä parantaa likiarvon tarkkuutta

30

matti.tedre@cs.joensuu.fi



## Tiedon lukeminen

- Pyydä käyttäjältä pizzan halkaisija  

```
# halk: tuoreimman säilyttävä  
halk = input("Anna pizzan halkaisija (cm):")
```
- Pyydä käyttäjältä pizzan hinta  

```
# hinta: tuoreimman säilyttävä  
hintaa = input("Anna pizzan hinta (e):")
```

31

matti.tedre@cs.joensuu.fi



## Tiedon käsittely

- Laske pizzan ala (cm<sup>2</sup>)
- Kaava:  $Pizzan\ ala = \pi \cdot r^2$
- Koska käyttäjä antaa pizzan halkaisijan ja kaavassa tarvitaan sädettä, tarvitaan tilapäissäilö johon lasketaan pizzan säde  

```
# sade: tilapäissäilö  
sade = halk / 2.0
```

32

matti.tedre@cs.joensuu.fi





## Tiedon käsittely (jatk.)

- Koska käyttäjä antaa tiedon senttimetreinä ja tuloksena ei haluta neliösenttimetrejä vaan neliömetrejä, lasketaan montako metriä säde on (muutetaan senttimetreistä metreiksi):

```
sade = sade / 100
```

- Lasketaan ala

```
# ala: tilapäissäilö  
ala = pii * sade * sade
```

33

matti.tedre@cs.joensuu.fi



## Tiedon käsittely (jatk.)

- Nyt tiedetään siis pizzan pinta-ala neliömetreinä
- Tarvitsee vielä laskea hinta per neliometri

```
# hinta_per_m2: tilapäissäilö  
hinta_per_m2 = hinta/ala
```

34

matti.tedre@cs.joensuu.fi



## Tiedon tulostus

- Tulostetaan käyttäjälle pizzan neliöhinta:

```
print "Pizza maksaa", hinta_per_m2, "e/m2"
```

35

matti.tedre@cs.joensuu.fi



## Ohjelmalistaus

```
# pii: kiintoarvo  
# halk, hinta: tuoreimman säilyttäjiä  
# sade, ala: tilapäissäilöjä  
# hinta_per_m2: tilapäissäilöjä  
pii = 3.1415  
halk = input("Anna pizzan halkaisija (cm):")  
hinta = input("Anna pizzan hinta (e):")  
sade = halk / 2.0  
sade = sade / 100 # senteistä metreiksi  
ala = pii * sade * sade  
hinta_per_m2 = hinta/ala  
print "Pizza maksaa", hinta_per_m2, "e/m2"
```

... Miksi luvut 2 ja 100 eivät ole kiintoarvoina?

36

matti.tedre@cs.joensuu.fi



## Ohjelman osat

```
# pii: kiintoarvo
# halk, hinta: tuoreimman säilyttäjiä
# sade, ala: tilapäissäilöjä
# hinta_per_cm2, hinta_per_m2: tilapäissäilöjä
pii = 3.1415
halk = input("Anna pizzan halkaisija (cm):")
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"
```

37

matti.tedre@cs.joensuu.fi



## Muuttujien roolit

- Muuttujan `pii` rooli tässä ohjelmassa on *kiintoarvo*



- Muuttujan `pii` arvoa ei muuteta sen jälkeen kun se alustetaan ohjelman alussa

38

matti.tedre@cs.joensuu.fi



## Muuttujien roolit

- Muuttujien `halk` ja `hinta` rooli tässä ohjelmassa on *tuoreimman säilyttäjä*



- Molempia muuttujia käytetään säilyttämään viimeksi syötettyä tietoa (vaikkakaan tässä ohjelmassa niiden arvo ei enää muutu)

39

matti.tedre@cs.joensuu.fi



## Muuttujien roolit

- Muuttujan `sade` rooli on *tilapäissäilö*



- Muuttujaa `sade` käytetään ohjelman selventämiseen, mutta ohjelma voitaisiin myös tehdä ilman tilapäissäilöä

40

matti.tedre@cs.joensuu.fi



## Muuttujien roolit

- Myös muuttujat `a1a` ja `hinta_per_m2` ovat *tilapäissäilöjä*



- Näitä muuttujia käytetään ohjelman selventämiseen, mutta ohjelma voitaisiin myös tehdä ilman niitä (tosin ohjelmasta tulisi vaikealukuinen)

41

matti.tedre@cs.joensuu.fi



## Arvojen vertailu

42

matti.tedre@cs.joensuu.fi



## Ongelma Sepon ohjelmassa

```
# äänen nopeus: kiintoarvo
# aika: tuoreimman säilyttäjä
# etäisyys: tilapäissäilö
aänen_nopeus = 340.0
aika = input("Anna kulunut aika:")
etaisyys = aika * aänen_nopeus
etaisyys = etaisyys / 1000
print "Salama löi ", etaisyys, "km:n päässä"
print "(C) Seppo 2006"
```

- Säättökija Seppo huomaa että etäisyyksiä laskevalle ohjelmalle voi antaa negatiivisia aika-arvoja

43

matti.tedre@cs.joensuu.fi



## Ohjelmaesimerkki (jatk)

- Seppo haluaa muuttaa ohjelmaa niin että ohjelma laskee etäisyyden vain positiivisilla arvoilla.
- Ohjelmassa täytyy siis kysyä
  - "Onko käyttäjän syöttämä `aika` yli nolla?"
- Ohjelman ei pitäisi laskea mitään mikäli aika on nolla tai alle.
  - Tarvitaan *ehtolause*

44

matti.tedre@cs.joensuu.fi



## Vertailulausekkeet ja ehtolause

- Samaa tyyppiä olevia arvoja voidaan vertailla keskenään *vertailulausekkeella*
- Numerot ja numeroarvoja säilyttävät muuttujat ovat samaa tyyppiä keskenään
- Vertailujen perusteella ohjelma voi valita mitä kohtia ohjelmasta suoritetaan tai jätetään suorittamatta (*ehtolause*)

45

matti.tedre@cs.joensuu.fi



## Operaattorit

- Vertailussa käytettävät *operaattorit* ovat hyvin samanlaisia kuin matematiikassa:

Pythonissa	Matematiikassa
<	<
<=	≤
==	=
>=	≥
>	>
!=	≠

46

matti.tedre@cs.joensuu.fi



## Vertailut

- Vertailut ovat aina joko *tosia* tai *epätosia*
- Esimerkivertailuja:

$5 < 7$  – tosi (on tosi, sillä 5 on pienempi kuin 7)

$5 > 7$  – epätosi (ei ole totta)

$5 == 7$  – epätosi (5 ei ole yhtä kuin 7)

$5 != 7$  – tosi (sillä 5 on erisuuri kuin 7)

47

matti.tedre@cs.joensuu.fi



## Vertailuja

- Vertailuissa on yleensä mukana yksi tai useampia muuttujia:
- Esimerkivertailuja (asetetaan ensiksi muuttujan arvo)

`hinta = 15`

`hinta < 10` – epätosi: (15 ei ole alle 10)

`hinta == 15` – tosi

`hinta-3 == 12` – tosi

48

matti.tedre@cs.joensuu.fi



## Vertailuja: tosia vai epätosia?

```
kirja1_sivuja = 112
kirja2_sivuja = 151

kirja1_sivuja < kirja2_sivuja

kirja2_sivuja > kirja1_sivuja

kirja1_sivuja != kirja2_sivuja
```

49

matti.tedre@cs.joensuu.fi



## Vertailuja: tosia vai epätosia?

```
benskulutus = 6.8

benskulutus < 8.8

benskulutus + 3.2 < 8.8

benskulutus * 2 < 8.8
```

50

matti.tedre@cs.joensuu.fi



## Ehtolauseet

- Vertailuja käytetään usein *ehtolauseissa*.
- Ehtolauseet alkavat sanalla **if** jonka perässä on vertailulauseke.
- Ehtolauseen perässä olevat *sisennetyt* rivit suoritetaan vain jos vertailuehto toteutuu
- Esimerkiksi korjattu Sepon sääohjelma:

```
if aika > 0:
    etaisyyys = aika * aanen_nopeus
    etaisyyys = etaisyyys / 1000
51 print "Salama löi ", etaisyyys, "km:n päässä"
```

matti.tedre@cs.joensuu.fi



## Ehtolauseet (2)

- Ehtolause siis kysyy: “*Onko vertailu totta?*”

```
if aika > 0:
    etaisyyys = aika * aanen_nopeus
    etaisyyys = etaisyyys / 1000
    print "Salama löi ", etaisyyys, "km:n päässä"
    print "(C) Seppo 2006"
```

- Ehtolause kysyy: “onko **aika** enemmän kuin 0?”
- Sisennetty osuus ohjelmasta suoritetaan vain *mikäli vertailu on totta*, eli vain jos **aika > 0**

52

matti.tedre@cs.joensuu.fi



## Ehtolauseet (3)

```
if aika > 0:
    etaisyyys = aika * aanen_nopeus
    etaisyyys = etaisyyys / 1000
    print "Salama löi ", etaisyyys, "km:n päässä"
print "(C) Seppo 2006"
```

- Ylläoleva voidaan lukea:
  - “Jos aika on enemmän kuin nolla, suorita seuraavat kolme riviä”.
- Ohjelman suoritus jatkuu riviltä ilman sisennystä, suoritettiin sisennetty osa tai ei

53

matti.tedre@cs.joensuu.fi



## Sepon ohjelma (versio 2)

```
# äänen nopeus: kiintoarvo
# aika: tuoreimman säilyttäjä
# etäisyys: tilapäissäilö
aanen_nopeus = 340.0
aika = input("Anna kulunut aika:")
if aika>0:
    etaisyyys = aika * aanen_nopeus
    etaisyyys = etaisyyys / 1000
    print "Salama löi ", etaisyyys, "km:n päässä"
print "(C) Seppo 2006"
```

- Nyt käyttäjän antamaa syötettä tutkitaan ennen kuin tulos lasketaan, ja virheellisellä syötteellä ei tulosteta muuta kuin copyright-tiedot

54

matti.tedre@cs.joensuu.fi



## Ohjelmaesimerkki (jatk)

- Uusi ongelma:
  - Jos käyttäjä antaa ajan joka on nolla tai alle, ohjelman suoritus vain pysähtyy.
- Seppo haluaisi että mikäli käyttäjä antaa virheellistä tietoa, ohjelma kertoo siitä käyttäjälle
- Voitaisiin tehdä kaksi ehtolauseetta...

55

matti.tedre@cs.joensuu.fi



## Kömpelö ratkaisu

```
# äänen nopeus: kiintoarvo
# aika: tuoreimman säilyttäjä
# etäisyys: tilapäissäilö
aanen_nopeus = 340.0
aika = input("Anna kulunut aika:")
if aika<=0:
    print "Ajan täytyy olla yli 0s."
if aika>0:
    etaisyyys = aika * aanen_nopeus
    etaisyyys = etaisyyys / 1000
    print "Salama löi ", etaisyyys, "km:n päässä"
print "(C) Seppo 2006"
```

- Monimutkaisemmissa tapauksissa usein virhealtis tapa
- ... on olemassa parempikin tapa tehdä

56

matti.tedre@cs.joensuu.fi



## Ehtolauseiden haarat

- Ehtolauseen jälkeinen sisennetty osa siis suoritetaan ainoastaan jos ehtolauseen vertailulauseke oli totta
- Ehtolauseeseen voidaan myös liittää `else`-haara joka suoritetaan “*kaikissa muissa tapauksissa*”

57

matti.tedre@cs.joensuu.fi



## Ohjelmaesimerkki

```
# äänen nopeus: kiintoarvo
# aika: tuoreimman säilyttäjä
# etäisyys: tilapäissäilö
aänen_nopeus = 340.0
aika = input("Anna kulunut aika:")
if aika>0:
    etaisyyys = aika * aänen_nopeus
    etaisyyys = etaisyyys / 1000
    print "Salama löi ", etaisyyys, "km:n päässä"
else:
    print "Ajan täytyy olla yli 0s."
print "(C) Seppo 2006"
```

58

matti.tedre@cs.joensuu.fi



## Mikä “haara”?

- Ohjelman suoritus “haarautuu” vertailun mukaan. Toiminta:
  1. Tehdään vertailu: “Onko aika nolla tai alle?”
  2. Mikäli vertailu on tosi (*kyllä*), suoritetaan kohta 3. mutta mikäli vertailu on epätosi (*ei*), suoritetaan kohta 4.
  3. Tulosta virheilmoitus
  4. Laske ja tulosta vastaus

59

matti.tedre@cs.joensuu.fi



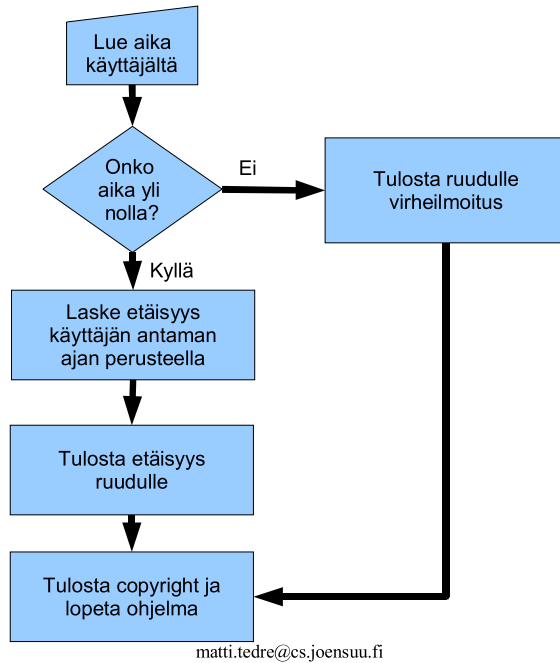
## Suorituksen haarautuminen

- Ohjelman suorituksen haarautumista voidaan kuvata myös piirroksella

60

matti.tedre@cs.joensuu.fi





61

## Valmis ohjelma

```

# äänen nopeus: kiintoarvo
# aika: tuoreimman säilyttäjä
# etäisyys: tilapäissäilö
aänen_nopeus = 340.0
aika = input("Anna kulunut aika:")
if aika>0:
    etaisyyys = aika * aänen_nopeus
    etaisyyys = etaisyyys / 1000
    print "Salama löi ", etaisyyys, "km:n päässä"
else:
    print "Ajan täytyy olla yli 0s."
print "(C) Seppo 2006"
  
```

62

## Esimerkkiohjelma

- Matemaatikko Maisa on kadottanut taskulaskimen ja tarvitsee ohjelman joka laskee jakolaskuja
- Ohjelman pitää pyytää käyttäjältä jakaja ja jaettava, ja laskea lukujen osamäärä
- Ongelma: nolllalla ei saa jakaa!

63

## Esimerkkiohjelma

- 1)Pyydä käyttäjältä jakaja
- 2)Pyydä käyttäjältä jaettava
- 3)Mikäli jakaja on nolla, tulosta virheilmoitus
- 4)Muussa tapauksessa tulosta lukujen osamäärä

64



## Esimerkkiohjelma (ehtolause)

- Jakolaskuohjelma saa suorittaa jakolaskun vain jos jakaja on *erisuuri kuin* nolla eli `jakaja != 0`

```
# jaettava, jakaja: tuoreimman säilyttäjiä
jaettava = input("Anna jaettava:")
jakaja = input("Anna jakaja:")
if jakaja != 0:
    print "Osamäärä on", jaettava / jakaja
```

65

matti.tedre@cs.joensuu.fi



## Esimerkki (jatk)

```
# jaettava, jakaja: tuoreimman säilyttäjiä
jaettava = input("Anna jaettava:")
jakaja = input("Anna jakaja:")
if jakaja != 0:
    print "Osamäärä on", jaettava / jakaja
else:
    print "Jakaja ei saa olla nolla"
```

- Lue: "Jos jakaja on erisuuri kuin nolla, tulosta osamäärä. Muussa tapauksessa tulosta virheilmoitus."

66

matti.tedre@cs.joensuu.fi



## Pizzalaskuri

- Ohjelmassa joka laskee pizzojen hintoja on myös virhe!
- Mikäli käyttäjä antaa pizzan halkaisijaksi nolla, on pizzan ala nolla, ja rivillä...  
`hinta_per_m2 = hinta/ala`
- ...yritetään jakaa nollalla!
- Ohjelman suoritus keskeytyy virheilmoitukseen

67

matti.tedre@cs.joensuu.fi



## Virheilmoitus

```
>>>
```

```
Anna pizzan halkaisija (cm):0
Anna pizzan hinta (e):25
```

```
Traceback (most recent call last):
  File "pizza.py",
    line 14, in -toplevel-
      hinta_per_m2 = hinta/ala
ZeroDivisionError: float division
>>>
```

68

matti.tedre@cs.joensuu.fi



## Korjattu ohjelmalistaus

```
# pii: kiintoarvo
# halk, hinta: tuoreimman säilyttäjiä
# sade, ala, hinta_per: tilapäissäilöjä
pii = 3.1415
halk = input("Anna pizzan halkaisija (cm):")
hinta = input("Anna pizzan hinta (e):")
if halk <= 0:
    print "Anna pizzan halkaisijaksi",
    print "positiivinen luku"
else:
    sade = halk / 2.0
    sade = sade / 100
    ala = pii * sade * sade
    hinta_per_m2 = hinta/ala
    print "Pizza maksaa", hinta_per_m2, "e/m2"
```

69

matti.tedre@cs.joensuu.fi

JOENSUUN  
YLIOPISTO

## Sepon ohjelma tekee analyysin

- Säättökija Seppo haluaa vielä kokeilla ohjelmaansa muutosta jossa ohjelma analysoisi tuliko salama kohdalle, lähelle, vai kauas.
- Kaikki salamat pitää luokitella siis etäisyyden perusteella:
  - Virheelliset syötteet
  - Kohdalle osuneet
  - Lähelle osuneet
  - Kauas osuneet

70

matti.tedre@cs.joensuu.fi

JOENSUUN  
YLIOPISTO

## Salama-analyysi

- Seppo voi käyttää tähän ehtolauseiden ketjua:
  - Jos etaisyys<0:
    - Tulosta "Virheellinen syöte! Anna positiivinen luku"
  - Muussa tapauksessa, jos etaisyys==0:
    - Tulosta "Osui kohdalle. Kävikö pahasti?"
  - Muussa tapauksessa, jos etaisyys <= lähellä:
    - Tulosta "Osui lähelle."
  - Muussa tapauksessa:
    - Tulosta "Osui kauas."

71

matti.tedre@cs.joensuu.fi

JOENSUUN  
YLIOPISTO

## Salama-analyysi

- Jos etaisyys<0:
    - Tulosta "Anna positiivinen luku"
  - Muussa tapauksessa, jos etaisyys==0:
    - Tulosta "Osui kohdalle. Kävikö pahasti?"
  - Muussa tapauksessa, jos etaisyys <= lähellä:
    - Tulosta "Osui lähelle."
  - Muussa tapauksessa:
    - Tulosta "Osui kauas."
- ```
if etaisyys<0:
    print "Anna pos..."
elif etaisyys==0:
    print "Osui kohdalle"
elif etaisyys<=lahella:
    print "Osui lähelle"
else:
    print "Osui kauas"
```

72

matti.tedre@cs.joensuu.fi

JOENSUUN  
YLIOPISTO

# Salama-analyysiohjelma

```
aanen_nopeus = 340.0
lahella = 1.5 # 1.5 km tai alle on lähellä
aika = input("Anna kulunut aika:")
etaisyys = aika * aanen_nopeus
etaisyys = etaisyys / 1000
if etaisyys<0:
    print "Anna 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"
```