### 1. LEKCIJA

##### 1.1. MAINĪGIE

Mainīgais ir koda sastāvdaļa, kurš apzīmē kaut kādu informāciju. Ruby valodā mainīgajiem datu tipi tiek piešķirti automātiski, taču reizēm ir nepieciešama to konvertācija uz citu tipu. Mainīgajiem gan ir fiksēti tipi – string, integer, decimal, boolean un citi, kā arī ir iespējami citi tipi un jebkāds cits datu kopums.

#šis ir komentārs tekstā

#mainīgais string - teksta rinda

string **=** "teksts, garš vai īss"

#mainīgais integer (fixnum) - skaitlis

integer **=** 6

#mainīgais decimal (float) - skaitlis ar komatu

decimal **=** 5.5

#mainigas boolean – loģiskais – tikai divas iespējas – patiess vai nepatiess

boolean **=** **true**

boolean **=** **false**

#pārvērst uz string

string **=** integer**.**to\_s

#pārvērst uz integer

integer **=** decimal**.**to\_i

#pārvērst uz decimal

decimal **=** integer**.**to\_f

##### 1.2. Elementāras darbības

Programmēšanas valodās ir iespējams veikt elementāru darbību aprēķināšanu

a **=** 2 **+** 2 # a vērtība ir 4 (2+2)

b **=** a **+** 2 # b vērtība ir 6 (4+2)

a **=** a **+** 2 # a vērtība ir 6 (4+2)

a **=** a **\*** 2 # a vērtība ir 12 (6\*2)

c **=** a **+** b # b vērtība ir 18 (12+6)

a **=** 3 # a vērtība ir 3

b **=** 2 # b vērtība ir 2

c **=** a **\*\*** b # c vērtība ir 9 (3 kāpināts 2. pakāpē - 3\*3)

##### 1.3. IZDRUKĀŠANA

Lai izvadītu kaut kādu rezultātu komandrindā ruby valodā ir komanda puts. Šai gadījumā:

puts "izvadāmais teksts" # izvadīs - izvadāmais teksts

a **=** 55 # ja nepieciešams varam izvadīt arī mainīgo rezultātus

puts a # izvadīs - izvadāmais mainīgais

puts "rezultāts: "**+**a**.**to\_s #izvadīs - rezultāts: 55

##### 1.4. mazas programmas

Piemērs mazai programmai – taisnstūra perimetra aprēķināšana, ja viena mala ir 5cm un otra 6cm:

# mainīgie

viena\_mala **=** 5

otra\_mala **=** 6

# pilns perimetrs ir viena mala reiz divi plus otra mala reiz divi

perimetrs **=** 2 **\*** viena\_mala **+** 2 **\*** otra\_mala

puts "perimetrs ir "**+**perimetrs**.**to\_s # izvadīs - perimeters ir 22

Pitagora teorēmas realizācija programmā (**Pitagora teorēma:** Pitagora teorēma ir sakarība starp taisnleņķa trijstūra malu garumiem un tā hipotenūzas garumu: ja taisnleņķa trijstūra katešu garumi ir a un b, bet hipotenūzas garums ir c, tad a2+b2=c2.) – pieņemsim , ka katetes ir 3cm un 5cm:

# mainīgie

kattete\_a **=** 3

kattete\_b **=** 5

# aprēķins hipotenūzai (mailai c)

hipotenuzas\_kvadrats **=** kattete\_a**\*\***2**+**kattete\_b**\*\***2

hipotenuza **=** Math**.**sqrt**(**hipotenuzas\_kvadrats**)**

puts "hipotenuza ir: "**+**hipotenuza**.**to\_s**+**" cm"

##### 1.5. Funkcijas

Programmēšanā bieži vien nākas izmantot vienu un to pašu koda gabalu vairākas reizes. Līdz ar to, funkcijas ir tas, kur mēs varam izveidot vairākas reizes „izsaucamus” koda gabalus. Piemērs funkcijai, kas veic aprēķinu taisnstūra laukumam (mala reiz mala):

**def** **laukuma\_aprekinasana** viena\_mala**,** otra\_mala

# funkcijas nosaukums ir laukuma\_aprekinasana

# tai būs nepieciešams padot divus mainigos

# pirmajam mainīgajam ko padosiet būs nosaukums "viena\_mala", otrajam "otra\_mala"

laukums **=** viena\_mala**\***otra\_mala

**return** laukums # ko atgriezt funkcijas rezultātam

**end**

laukums **=** laukuma\_aprekinasana 5**,** 6

puts "laukums ir "**+**laukums**.**to\_s

##### 1.6. if (nosacījuma) operators

Dažreiz ir nepieciešams izveidot tādu vai savādāku nosacījumu pēc kura strādā programma. Piemērs: ja Jānis ir salasījis vairāk pa 5 āboliem, programmai jāpasaka Jānim, ka pietiek lasīt, bet ja ne, tad jālasa vēl.

jana\_abolu\_daudzums **=** 5

**if** jana\_abolu\_daudzums **>** 5

puts "pieteik lasīt"

**else**

puts "jālasa vēl"

**end**

If operators var eksistēt arī viens pats, piemēram, tikai jāpasaka Jānim tad kad pieteiks lasīt:

jana\_abolu\_daudzums **=** 5

**if** jana\_abolu\_daudzums **>** 5

puts "pieteik lasīt"

**end**

Kā arī if operators var ietvert sevī daudz vairāk „zaru”, jebšu nosacījumu dažādiem gadījumiem ar elsif.

jana\_abolu\_daudzums **=** 5

**if** jana\_abolu\_daudzums **>** 5

puts "pieteik lasīt"

**elsif** jana\_abolu\_daudzums **==** 5

puts "janim ir 5"

**else**

puts "janim ir maz"

**end**

##### 1.7. while – cikliskais operators

While ir operators, kas izpildās, kamēr nosacījums ir patiess. Piemērs, ja jānim ir mazāk pa pieciem āboliem, katru reizi, kā jānis paceļ ābolu, viņam pasaka, ka jālasa vēl:

# mainigie

jana\_abolu\_daudzums **=** 0 # janim no sakuma nav abolu

**while** jana\_abolu\_daudzums **<** 5 # kamer janim abolu daudzums ir mazaks pa pieci

puts "jālasa vēl" # pasakam jānim, ka jālasa vēl

jana\_abolu\_daudzums **=** jana\_abolu\_daudzums **+** 1 # pieliekam janim pie abolu daudzuma vēl vienu

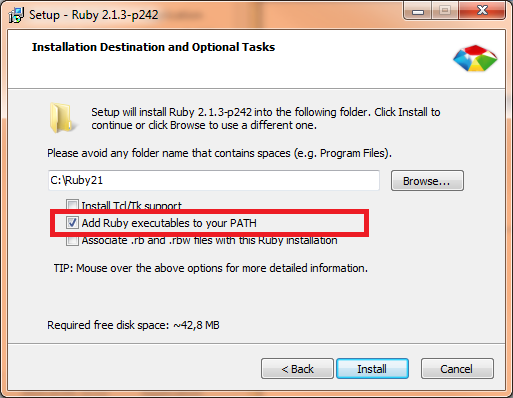
**end**

puts "pietiek lasīt" # pasakam jānim, ka viņam ir pieci un pietiks lasīt

##### 1.8. programmu izpilde uz Windows vides

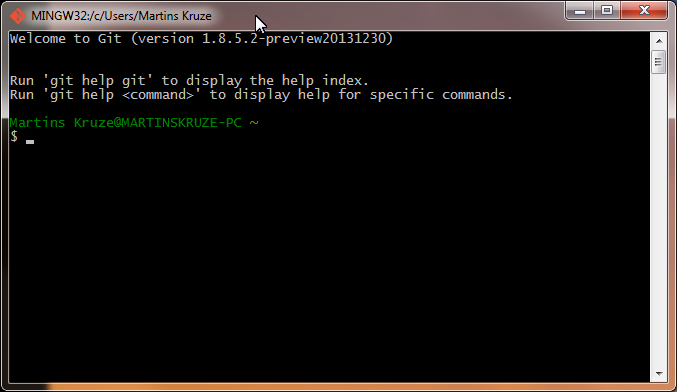
Lai programma varētu izpildītes, tad nepieciešams sekojoši sagatavot vidi:

1. (<http://rubyinstaller.org/downloads/>) **Ruby valodas kompilators** (instalējot 4. solī nepieciešams ruby valodu „Add RUBY executables to PATH”)!!!



1. (<http://git-scm.com/download/win>) **Git komandrinda**

**Palaižam GIT Bash – komandrindu:**



„Aizejam” uz folderi, kur atrodas programmas kods, kuru mēs esam sarakstījuši. Manā gadījumā tas ir mans lietotājs (vieta kurā atrodas git komandrinda jau atverot) => repos => iespeju\_tilts => 01\_lekcija\_30\_09\_2014 => kods. Līdz tādam ceļam var aiziet ar komandu cd repos/iespeju\_tilts/01\_lekcija\_30\_09\_2014/kods/, kur **cd** ir change directory, jebšu *mainīt mapīti*. Lai spētu, „izpildīt komandu”. Jāpadod RUBY kompilatoram koda faila nosaukums, šajā gadījumā fails „pitagors.rb”.

Piemērs:

Martins Kruze@MARTINSKRUZE-PC ~

$ **cd repos/iespeju\_tilts/01\_lekcija\_30\_09\_2014/kods/**

Martins Kruze@MARTINSKRUZE-PC ~/repos/iespeju\_tilts/01\_lekcija\_30\_09\_2014/kods (master)

$ **ruby pitagors.rb**

hipotenuza ir: 5.830951894845301 cm

