

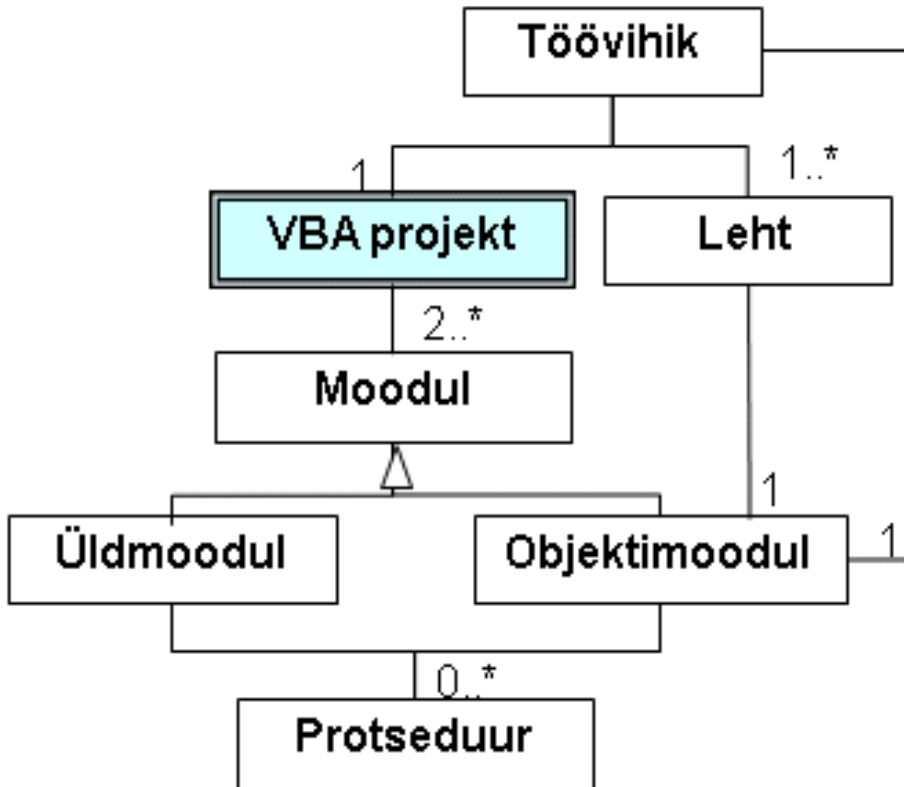
Rakenduste loomise ja programmeerimise alused

VBA. Sissejuhatus

Keskkonna seadistamine VBA programmide kasutamiseks *MS Office's*

- ▶ **VBA**-d kasutavad *MS Office* failid (alates *MS Office 2007*), tuleb salvestada koos VBA projektiga st. *Macro-enabled* vormingus (laiend lõpeb tähega *m*: *.xlsm*, *.docm*, *.pptm* jm)
- ▶ *MS Office 2003* ja varasemad failid salvestatakse alati VBA projektiga st. *Macro-enabled* vormingus
- ▶ **VBA** käivitamine ja kasutamine toimub vahekaardi **Developer** korralduste abil. Vaikimisi on vahekaart peidetud, näitamiseks vt. **Office** menüü – **Options ...**
- ▶ MSO 2007: Trust Center – Trust Center Settings – Macro Settings
 - ▶ märkida märkeruut *Trust Access to the VBA project object model*
 - ▶ valida suvanditest sobiv turvalisuse tase:
Enable all macros / Disable all macros with notification / ...
- ▶ Faili avamisel makrode kasutamise lubamiseks valida turvalisuse tase: *Enable macros*

VBA projekti struktuur



VBA projektis on moodulid, kuhu saab sisestada protseduure.

- objektimoodulid
- üldmoodulid.

Oma objektimoodul on igal lehel ja töövihikul. Lehemoodul luuakse ja eemaldatakse koos lehega.

Objektimoodulisse salvestatakse protseduurid, mis on seotud antud objektiga.

Üldmoodulisse salvestatakse üldise iseloomuga protseduure, mis võivad olla seotud mitme lehega. Üldmooduleid lisatakse korraldusega **Insert/Module**, nende arv ei ole piiratud.

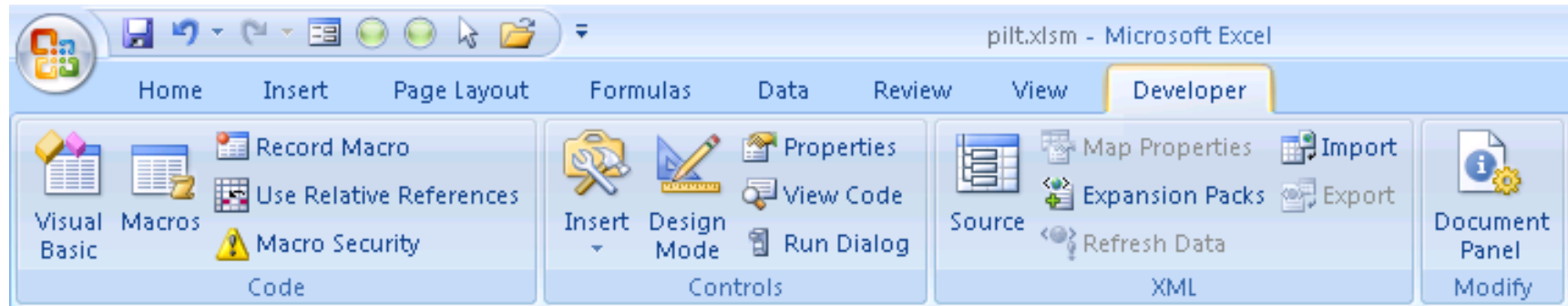
VBA akna aktiveerimine

VBA programmid salvestatakse VBA projekti moodulitesse Visual Basic redaktori (VBE – Visual Basic Editor) abil.

Redaktori aktiveerimiseks saab kasutada erinevaid viise:

- ▶ käsku (nuppu) vahekaardil **Developer**.
- ▶ klahvidega **Alt+F11**.
- ▶ klõpsates hiire parempoolse nupuga töölehelipikut ja valides ilmuvast menüüst korralduse **View Code** (see viib aktiivse lehe moodulisse)

Kui Visual Basicu redaktorit on antud seansis juba kasutatud, asub tegumiribal vastav ikoon, mis võimaldab kohe pääseda redaktori keskkonda.



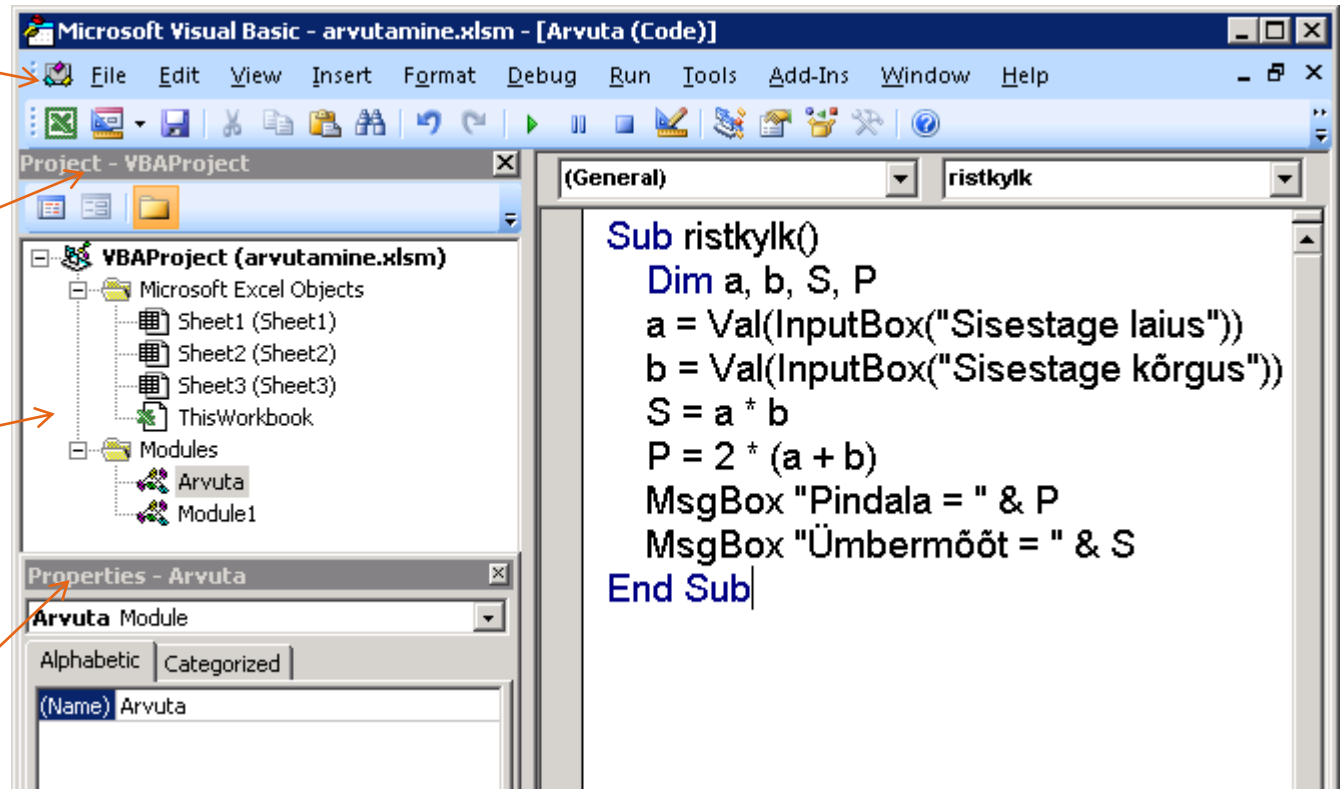
VBA projekti struktuur ja redaktori kasutajaliides

menüü ja standardriistariba

projektihalduri aken
Project Explorer

moodulite puu iga projekti (avatud töövihiku) jaoks

Põhiobjektide omaduste aken: *Properties Window*



VBA keskkond. Projektihalduriaken

- ▶ **Projektihalduriaken** (*Project Explorer*) võimaldab operatsioone programmimoodulitega (lisamine, avamine/aktiveerimine, kustutamine), selles on näha kõikide selles rakenduses avatud dokumentidega seotud standard-, klassi- ja objektimoodulid.
- ▶ Projektihalduriakna avamiseks
 - ▶ ***View – Project Explorer***
 - ▶ ***Ctrl+R***.
- ▶ **Mooduli lisamine:** peamenüüst või hüpikmenüüst: ***Insert*** – mooduli tüüp.
- ▶ **Valitud mooduli avamine:**
 - ▶ topeltklõps mooduli nimel,
 - ▶ peamenüüst ***View – Code***,
 - ▶ hüpikmenüüst ***View Code*** või
 - ▶ funktsiooniklahv ***F7***.
- ▶ **Valitud mooduli kustutamine:**
 - ▶ ***File – Remove...***
 - ▶ hüpikmenüüst ***Remove***.

VBA keskkond. Koodiaken

- ▶ Koodiaken (üks iga mooduli kohta) sisaldab protseduuride tekste ja kirjeldusi, on mõeldud programmitekstide lisamiseks ja muutmiseks,
- ▶ Koodiakna avamiseks topeltklõps või **View – Code** või funktsiooniklahv **F7**.
- ▶ Iga mooduli jaoks võib olla avatud oma koodiaken, nii on mugav programmitekste võrrelda ja vajadusel kopeerida
- ▶ Protseduure näidatakse koodiaknas kas ühekaupa (*Procedure View*) või korraga (*Full Module View*), valida nuppude abil akna allservas.
- ▶ Ripploendiboksid *Object* ja *Procedure* akna ülal servas võimaldavad valida sobiva objekti (objektimoodulis) ning objektiga seotud sündmuse (*Event*) või protseduuri (kui valitud objekt *General*).
- ▶ Võimalik lisada/eemaldada katkestuspunkte. Katkestusrežiimis saab selles aknas jälgida protseduuride täitmist.

VBA laused

- ▶ Iga VBA lihtlause peaks olema eraldi real.
- ▶ Kaht lauset ühel real peab eraldama : (*koolon*).
- ▶ Pikema lause jagamiseks mitmele reale pannakse poolituskohta _ (*tühik ja alakriips*).

VBA laused koosnevad:

- ▶ võtmesõnad (*Sub, Dim, End, If* jm)
- ▶ konstandid (arv, tekst jm.)
- ▶ nimed (muutujad, alamprotseduurid)
- ▶ avaldised (operatsioonid konstantide ja muutujatega, sh. pöördumine funktsiooni(de) poole)

VBA programm

- ▶ VBA programm määrab tegevusi andmete ja objektidega:
 - ▶ **andmed:** arvud, tekstid jm
 - ▶ **objektid:**
 - ▶ Exceli töövihikud, töölehed, lahtrid
 - ▶ tekstidokumendid, tekstilõigud jm
 - ▶ graafilised kujundid rakendustes
- ▶ kahte liiki VBA protseduure:
 - ▶ **sub-protseduur** – võimalik määratleda suvalisi tegevusi suvaliste objektidega
(makro – parameetriteta sub-protseduur)
 - ▶ **funktsioon** – võimalik leida ainult üks väärtus (NB! tagastatav väärtus tuleb omistada funktsiooni nimele)

VBA-protseduuride tüüpstruktuur

Makro peaprotseduur. Parameetriteta **Sub**-protseduur

Sub nimi ()

laused ja kommentaarid

End Sub

Sub-tüüpi **alamprotseduur.** Võivad olla sisend- ja väljundparameetrid

Sub nimi ([parameetrid])

laused ja kommentaarid

End Sub

Pöördumine **Call**-lausega: **[Call]** nimi ([argumendid]) **NB!** nurksulgudes [] osa võib puududa!

Function-protseduur. Võivad olla sisendparameetrid

Function f_nimi([parameetrid])

laused ja kommentaarid

f_nimi = tagastatav väärtus

End Function

Pöördumine: funktsiooniviit **f_nimi** ([argumendid])

Programmide täitmisest I

- ▶ Protseduuride täitmist korraldab spetsiaalne programm – **VBA interpretaator** (translaator).
- ▶ Interpretaator eraldab protseduurile arvuti mälus täitmise ajaks **tööpiirkonnad: programmiplokk ja andmeplokk**.
- ▶ Programmiplokki tuuakse vastava protseduuri tekst ning kontrollitakse selle süntaksi õigsust.
- ▶ Kui on vigu, väljastatakse veateade ja täitmine katkestatakse.
- ▶ Andmeplokis eraldatakse mälupesad (väljad) protseduuri poolt kasutatavate muutujate väärtuste ajutiseks säilitamiseks.
- ▶ Interpretaator tõlgib protseduuri lausete kaupa masinkeelde, suunab täitmiseks protsessorile ja tagastab tulemused andmeplokki.

Programmide täitmisest II

- ▶ Programmi täitmise alustamine/lõpetamine VBA-aknas.
Menüüvalik **Run**:
- ▶ **Run (F5)** – protseduuri käivitamine, katkestusrežiimis ka jätkamine (**Continue**)
- ▶ **Break (Ctrl+Break)** – programmi täitmise katkestamine
- ▶ **Reset** – programmi täitmise lõpetamine (sobib katkestusrežiimis, kui soovitakse käivitada programm uuesti algusest)

Makrode käivitamine

- ▶ Käsuga **Macros** vahekaardil **Developer**. Ka Alt+F8

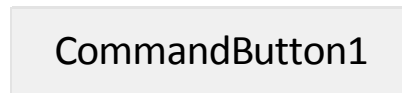
- ▶ **Forms** ja **ActiveX** käsunuppude abil.

Mõlemat tüüpi elementide loomise jaoks kasutatakse vahekaardi **Developer** grupi **Controls** käsku **Insert**

- ▶ Käsunupp Forms



- ▶ Käsunupp ActiveX



- ▶ **graafikaobjekti** abil

objektimenüüst *Assign Macro*



- ▶ Kiirklahvikorraldusega **Ctrl + täht** (näiteks Ctrl+q) määramiseks vt. Makros - Options

Makro sidumine graafikaobjektiga

- ▶ Kasutades joonestamise vahendeid joonistada töölehele suvaline kujund või importida mingi kujund.
- ▶ Klõpsata kujundit hiire parempoolse nupuga ning valida ilmuvast objektimenüüst korraldus **Assign Macro**
- ▶ Ilmuvast dialoogiboksist **Assign Macro** valida vajalik makro ja klõpsata nuppu **OK**

Kui makroga seotud graafikaobjekt on vaja teha aktiivseks, et muuta tema asukohta, suurust vm, peab klõpsamise ajal hoidma all klahvi **Ctrl**.

Start



Programmi sisestamine

- ▶ VBE kontrollib süntaksi õigsust lausete kaupa.
Kui lauses on süntaksivigu, kuvatakse üleminekul uuele reale vastav veateade, muudetakse teksti värvus punaseks ning paigutatakse kursor vea eeldatavale asukohale.
- ▶ VBE toetab objektide omaduste ja meetodite sisestamisel.
Kui peale objektiviita sisestada punkt, pakub redaktor antud klassi objektide omaduste ja meetodite loendi
- ▶ Teavet objektide, nende omaduste ja meetodite kohta saab objektisirvija abil (*Object Browser*)
- ▶ Teavet võtmesõna, omaduse, meetodi jmt. kohta: viia kursor vastava elemendi kohale ja vajutada klahvile **F1**.

Muutujad I

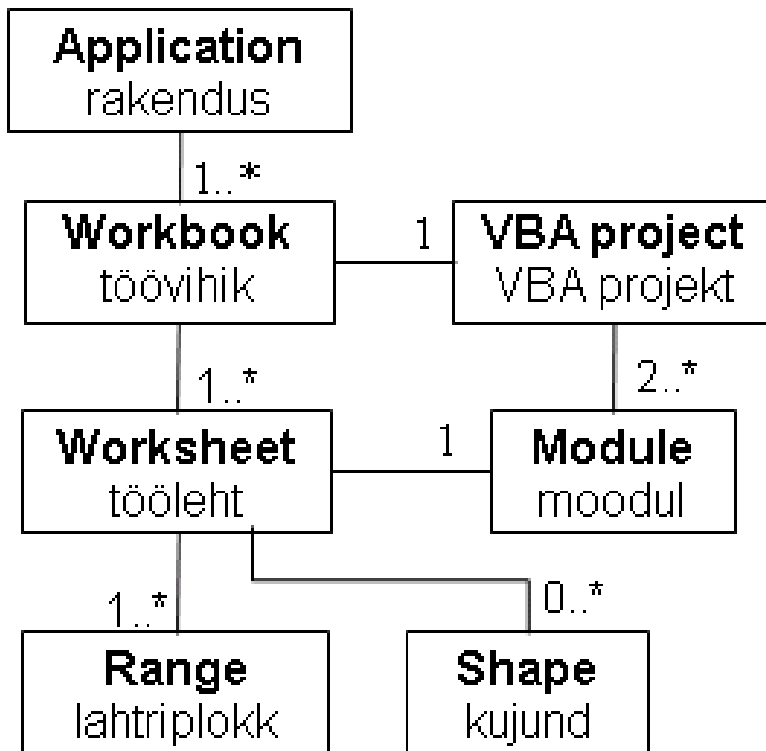
- ▶ **Muutuja** (*Variable*) on nimega mälupesa ehk väli
- ▶ Igal muutujal on **nimi** ja **skoop** (mõjupiirkond)
- ▶ Üldlevinud reeglid nimede määramisel:
 - ▶ üks täht
 - ▶ tähtede, numbrite ja allkriipsude jada, mis algab tähega.
- ▶ Skoop - muutuja mõjupiirkond: globaalne või lokaalne
 - ▶ **Globaalne** – kättesaadav ehk nähtav kõikidele
 - ▶ **Lokaalne** (kättesaadav ainult antud protseduurile)

Muutujate kirjeldamine (deklareerimine)

- ▶ Deklaratsiooniga määratakse muutuja:
 - ▶ nimi
 - ▶ tüüp
 - ▶ skoop ehk mõjupiirkond
 - ▶ väärtuste eluiga
 - ▶ Peamiselt deklareeritakse muutujad **Dim**-lausega:
Dim nimi [As tüüp] { , nimi [As tüüp] }
 - ▶ lihtmuutujate tüüp – võtmesõna **Integer, Long, Single, Double, String, Boolean, Date** või **Variant**
 - ▶ objektimuutujate tüüp esitatakse klassi nime abil:
Range, Shape, Worksheet.
- Kui tüüpi pole näidatud, siis võetakse selleks **Variant**.

Exceli objektide kasutamine VBA's

Rakenduse struktuur ja põhiobjektid



Exceli rakendus – omavahel seotud objektide kogum: töovihikud, töölehed, lahtriplokid, graafilised kujundid, ...

Objektitüübil ehk klassil on kindel nimi: *Workbook*, *Worksheet*, *Range*, *Shape* ...

Klassi nimesid kasutatakse viitamiseks objektidele:

Range(„lööke“), *Shapes*(“pall”)

VBA protseduur ei ole seotud konkreetse objektiga ning üks protseduur saab määrata tegevusi mitme objektiga

Exceli objektid. Nõuandenurk

- ▶ **Objektile** (lahtrile, lahtriplokile või graafikaobjektile) **nime andmiseks** valida see objekt, tippida nimi (tähed või numbrid või alakriips) nimekasti (varemiriba vasakus servas), vajutada ENTER
- ▶ Kui nimedeks sobivad tekstid on sisestatud kõrvallahtritesse, valida lahtriplokk – nimetatavad lahtrid ja sobivad nimed – ning kasutada korraldust:
Formulas – (Defined Names) – Create from Selection
- ▶ Töölehega seotud moodulilehele saab töölehe nimelipikul paremklõpsuga saadud objektimenüüst valides **View Code**
- ▶ Töölehega seotud moodulilehele kirjutatakse VBA protseduurid, milles kasutatakse sellel töölehel olevaid objekte.

Tööleht – klass Worksheet

Tööleht
nimi
ridade arv
veergude arv
laius
kõrgus
aktiivne lahter
...
aktiveerimine()
teisaldamine()
kopeerimine()
eemaldamine()
peitmine()
...

- ▶ VBA abil saab määrata tegevusi Exceli **töövihikutega** (avamine, sulgemine jm) ja **töölehtedega** (lisamine, kopeerimine, eemaldamine jm)
- ▶ Protseduurid salvestatakse töölehe moodulisse või üldmoodulisse.
- ▶ Salvestades töövihiku, saame säilitada nii rakenduse töö tulemused kui ka vastavad programmid.

Lahtrid ja lahtriplokid – klass Range

- ▶ Lahtriplokina ehk **Range**-objektina võib käsitleda töölehe lahtrite suvalist kogumit (ka üksiklahtrit)
- ▶ Viitamiseks sellele objektile on põhivariant: **Range(id)**
id (identifikaator) on lahtri või piirkonna nimi või aadress või muutuja, mille väärtuseks on lahtriploki nimi või aadress
Näited: **Range("A2")**, **Range("pindala")**, **Range("aeg")**
- ▶ Viitamiseks teise töölehe lahtritele, peab lahtriviidale eelnema viit lehele, näiteks:
Sheets("Hinnakirjad").Range("Arvutid")
Sheets("Maksud").Range("tulumaks")

Graafilised kujundid – klass Shape

Klassi **Shape** kuuluvad:

- ▶ Baaskujundid: sirgjoone lõik, riskülik, hulknurk, ovaal jm
- ▶ Kõik töölehel paiknevad objektid: joonised, imporditud pildid, ohjurid (käsunupud, kerimisribad, märkeruudud jm)
- ▶ Viitamiseks neile võib kasutada **Shapes (*nimi*)**, kus ***nimi*** on objekti ehk kujundi nimi.

Shapes(“Rectangle13”), Shapes (“ring”), Shapes (“pall”)

- ▶ Kui protseduur ei asu selle lehe moodulis, millel on kujund, peab kujundi viidale eelnema viit lehele.
 - ▶ Viit kujundile, kasutades konkreetse lehe nime Sheets (“Mäng”).Shapes(“pall”)
 - ▶ Viit kujundile aktiivsel töölehel ActiveSheet.Shapes(“Juku”)

Objektimuutuja

- ▶ Objekti saab **Set**-lause abil siduda muutujaga. See võimaldab viitasid lühendada. Tegemist on omalaadse omistamislausega.
- ▶ Objektile viitavate muutujate deklareerimisel on mõistlik määrata nende tüüp (objekti tüübina)
näiteks **Dim auto as Shape, prk as Range**
- ▶ **Set**-lause üldkuju:
Set muutuja = objekt
- ▶ Näited:
 - ▶ **Set auto = Shapes("auto")**
 - ▶ **Set J = Shapes("Juku")**
 - ▶ **Set prk = Range("Tabel")**

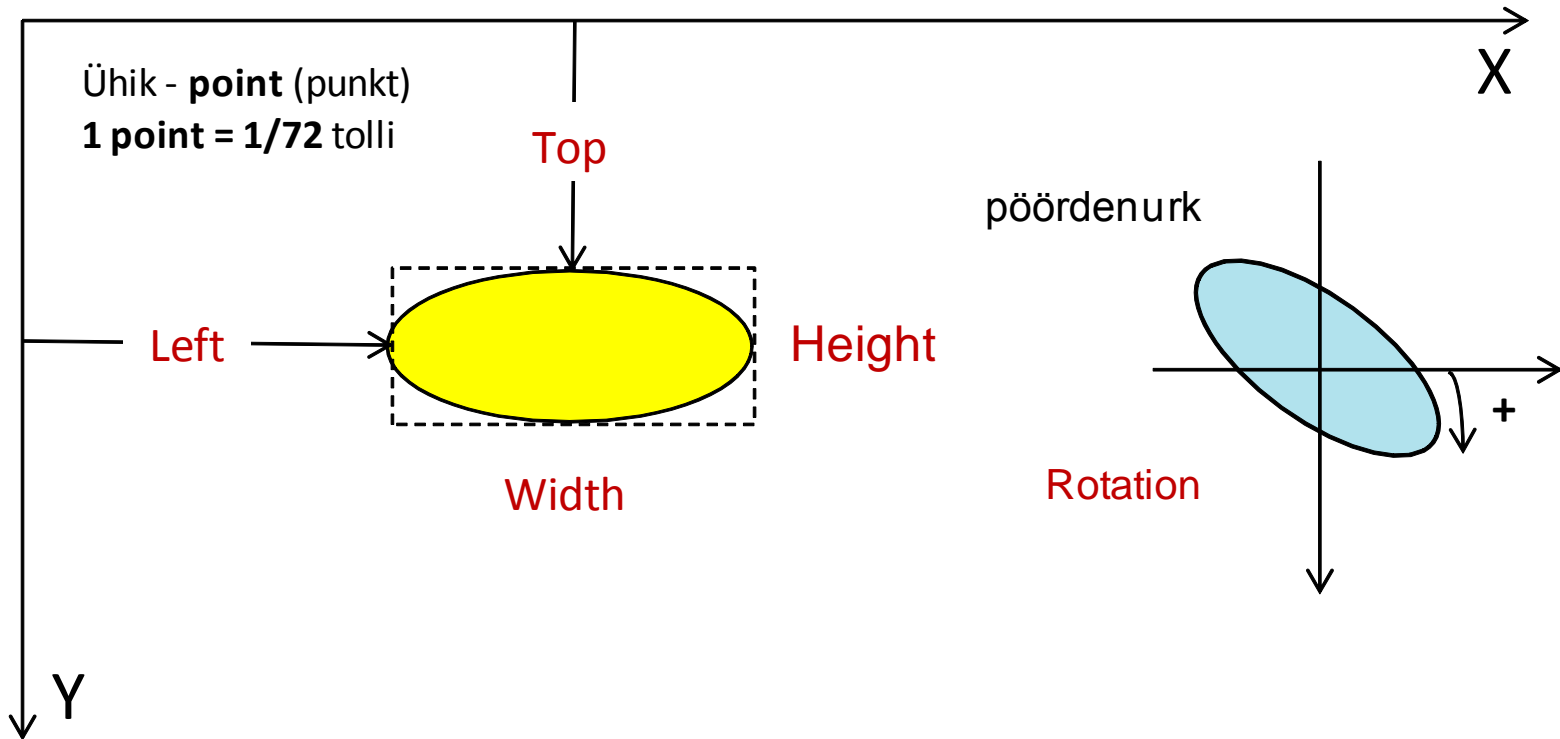
Töölehe koordinaadisüsteem ja mõõtühikud

Töölehe koordinaadisüsteem						
Nullpunkt - ülemises vasakpoolses nurgas						
	A	B	C	D	E	...
1	(0, 0)					X
2		Ühik - point (punkt)				
3	Y	1 point = $1/72'' = 2,54/72 \approx 0,035$ cm				
4		1 cm = $72 / 2,54$ punkti				
5						
...						

- ▶ Töölehte võib vaadelda kui ekraani piirkonda, kus tegutsevad graafikaobjektid
- ▶ Võrreldav Scratchi lavaga.

- ▶ Töölehe virtuaalne laius on ca 300 m ja kõrgus lausa 5 km.
- ▶ Koordinaatsüsteemi nullpunkt on töölehe ülemises vasakpoolses nurgas. Horisontaalne telg (**X**) on suunatud paremale, vertikaalne telg (**Y**) alla. Pikkusühikuks on trükindusest pärit üksus **point** (**punkt**), mis võrdub $1/72$ tolliga ehk ca 0,35 millimeetriga.

Shape-objekti geomeetrilised omadused



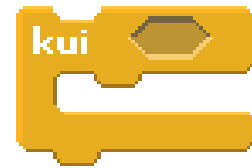
Valikulaused I

Valikulaused võimaldavad määrata tegevuste täitmist sõltuvalt etteantud tingimustest.

Valikulaused on liit- ehk struktuurilaused, võivad sisaldada teisi liht- ja/või liitlauseid.

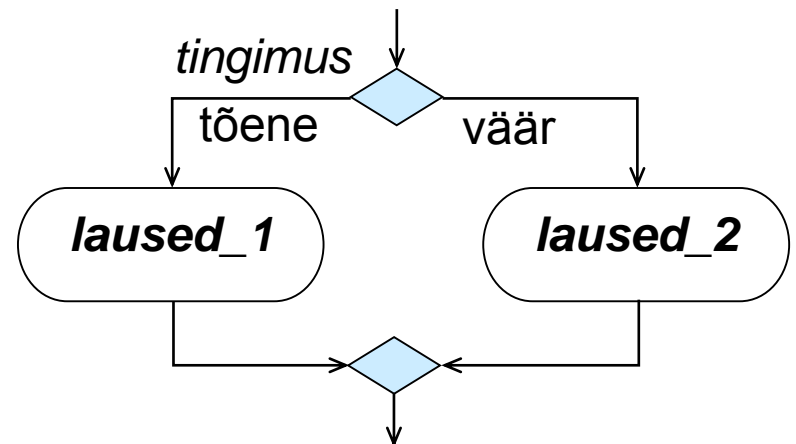
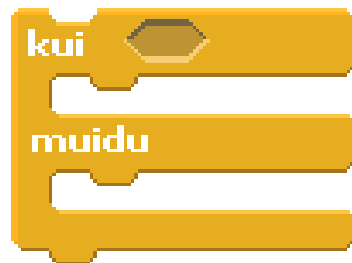
Üherealine If-lause

If tingimus Then laused_1 [Else laused_2]
(Kogu lause peab paiknema ühel real)



If-lause põhivariant (kahendvalik)

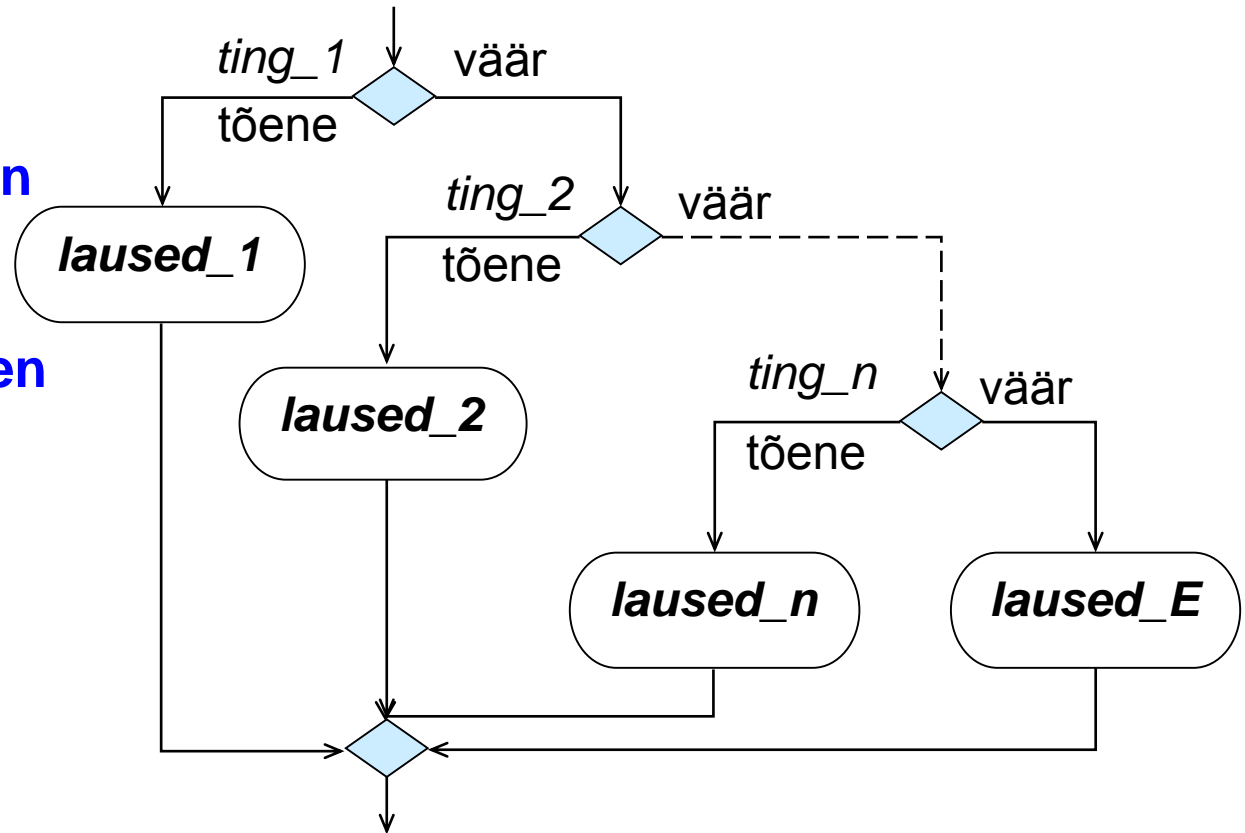
If tingimus Then
 laused_1
[Else
 laused_2]
End If



Valikulaused II (If-plokk)

If-plokk – üldjuht

```
If tingimus_1 Then  
  laused_1  
Elseif tingimus_2 Then  
  laused_2  
  ...  
Elseif tingimus_n Then  
  laused_n  
Else }  
  laused_E  
End If
```



Kordused ja korduslaused

VBAAs on korduste kirjeldamiseks mitu lauset:

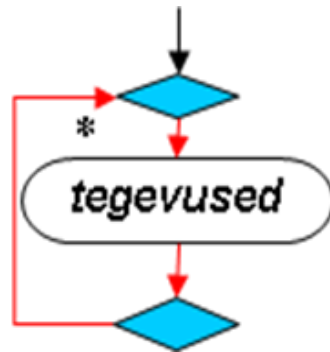
- ▶ Do...Loop-lause
- ▶ For-lause
- ▶ For Each-lause

Lausetel on mitu varianti

Do ... Loop-lause

Lõputu kordus

Do
laused
Loop



kordus
tegevused
lõpp kordus

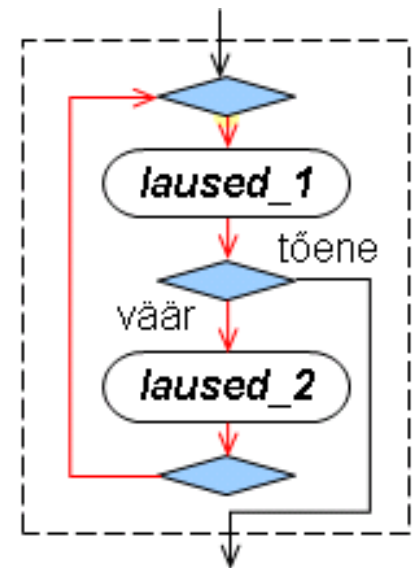


Lõputu kordus katkestusega

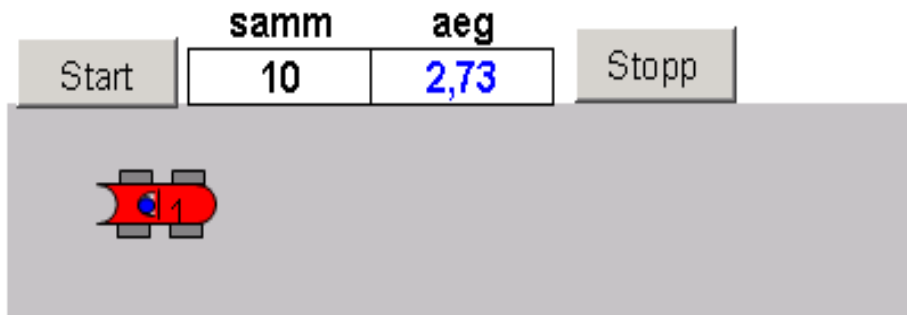
Do
laused_1
If *tingimus* Then Exit Do
laused_2
Loop



kordus
tegevused 1
kui *tingimus* siis välju
tegevused 2
lõpp kordus



Lõputu kordus katkestusega. Näide Auto 1



```
Sub Auto_1()  
  Dim auto As Shape, algaeg, h  
  Set auto = Shapes("auto")  
  h = Range("samm")  
  auto.Left = 0: algaeg = Timer()  
  Do  
    Range("aeg") = Timer() - algaeg  
    auto.IncrementLeft h / 2 + Rnd() * h  
    If auto.Left > Shapes("fin").Left Then Exit Do  
    paus 0.01  
  Loop  
  MsgBox "Olen kohal!"  
End Sub
```

Töölehel on kujundid nimedega **auto** ja **fin**. Programmi toimel „sõidab“ auto töölehe vasakust servast finišijooneni (**fin**).

Põhiosa tegevustest toimub **Do...Loop** korduses.

Igal kordamisel kirjutatakse lahtrisse **aeg** jooksev aeg.

Auto vasaku serva koordinaati muudetakse juhuarvu abil.

If-lauses kontrollitakse kas auto vasak serv on suurem finišijoone vasakust servast.

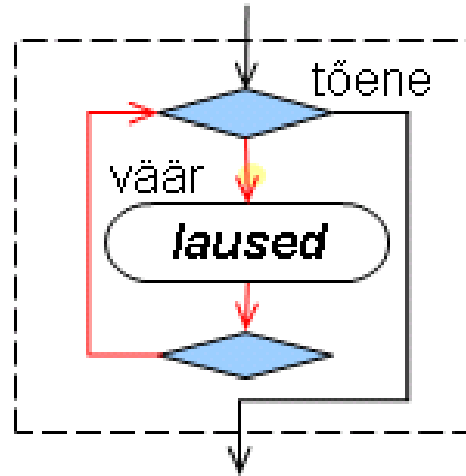
Autosõit

Do ... Loop-lause. Eelkontrolliga ja järelkontrolliga kordused I

Do Until *tingimus*

laused

Loop

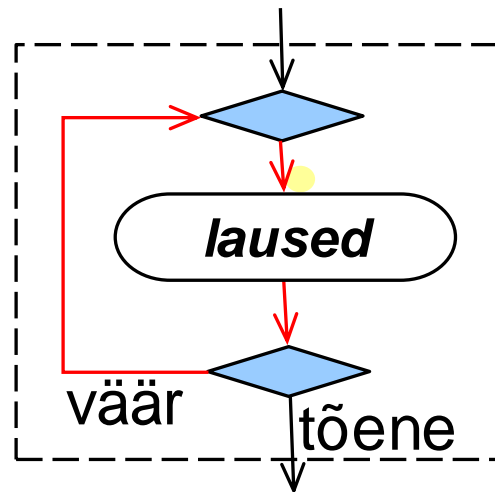


Kui **Do**-osalauses kasutatakse võtmesõna **Until**, toimub kordamine seni, kuni tingimus **saab tõeseks**.

Do

laused

Loop Until *tingimus*



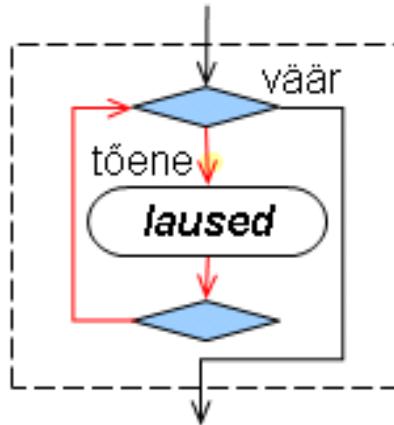
Järelkontrolliga lausetes täidetakse grupis olevaid lauseid vähemalt üks kord. Eelkontrolliga korduses võivad nad jääda ka täitmata, kui tingimus on **Until**-tüüpi korduses kohe **tõene**

Do ... Loop-lause. Eelkontrolliga ja järelkontrolliga kordused II

Do While *tingimus*

laused

Loop

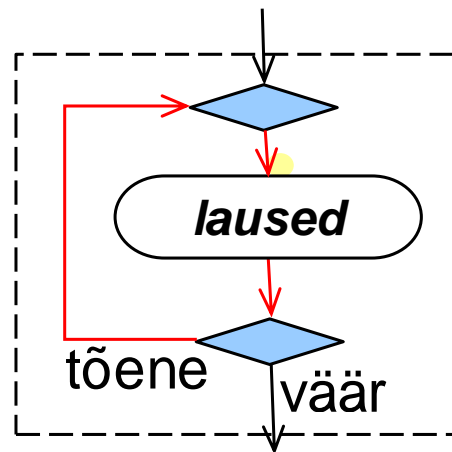


Kui **Do**-osaluses kasutatakse võtmesõna **While**, toimub kordus, kui tingimus on **tõene**.

Do

laused

Loop While *tingimus*



Järelkontrolliga lausetes täidetakse grupis olevaid lauseid vähemalt üks kord. Eelkontrolliga korduses võivad nad jääda ka täitmata, kui tingimus on kohe väär **While** korduse korral.

Eelkontrolliga Until-kordus. Näide Auto_2

```
Sub Auto_2()  
  Dim auto As Shape, algaeg, h  
  Set auto = Shapes("auto")  
  h = Range("samm")  
  auto.Left = 0: algaeg = Timer()  
  Do Until auto.Left >= Shapes("fin").Left  
    Range("aeg") = Timer() - algaeg  
    auto .IncrementLeft h / 2 + Rnd() * h  
    paus 0.01  
  Loop  
End Sub
```

Kui auto on juba sihtkohas, ei liigu see programmi käivitamisel paigast: tingimus on kohe tõene.

Järelkontrolliga While-kordus. Näide Auto_3

```
Sub Auto_3()  
  Dim auto As Shape, algaeg, h  
  Set auto = Shapes("auto")  
  h = Range("samm")  
  auto.Left = 0: algaeg = Timer()  
  Do  
    Range("aeg") = Timer() - algaeg  
    auto.IncrementLeft h / 2 + Rnd() * h  
    paus 0.01  
  Loop While auto.Left < Shapes("fin").Left  
End Sub
```

Kui auto on juba sihtkohas, teeb auto ühe sammu ja siis selgub, et tingimus on väär ja kordus lõpetatakse.



Autosõit

For ... Next kordus. Juhtmuutujaga kordus

For v = av1 **To** av2 [**Step** av3]

laused_1

[**Exit For**]

laused_2

Next v



v on juhtmuutuja, av1, av2 ja av3 võivad üldjuhul olla avaldised.

av1 – juhtmuutuja algväärtus,

av2 – lõppväärtus,

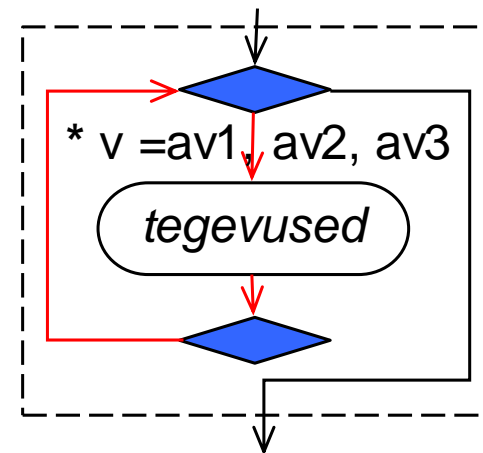
av3 – muutmise samm.

Kui samm ei ole antud, võetakse selle väärtuseks 1.

kordus v = a1..a2 [,a3]

tegevused

lõpp kordus



For...Next kordus. Juhtmuutujaga kordus

```
For jrk = 1 To mitu Step 1
```

```
.....
```

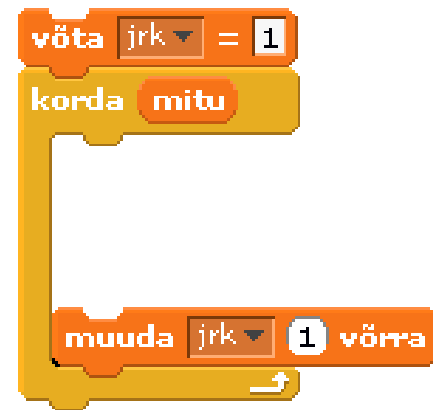
```
.....
```

```
Next jrk
```

Siin on tegemist samade tegevustega.

Korduse lõppedes on juhtmuutuja väärtus sammu võrra suurem lubatud lõppväärtusest.

```
jrk = 1  
Do Until jrk >= mitu  
.....  
  jrk = jrk + 1  
Loop
```



For ... Next-lause. Näide „Auto ringliiklus“

```
Sub Auto_4()  
  Dim auto As Shape, ringe, ring, algaeg  
  Set auto = Shapes("auto")  
  auto.Left = 0  
  ringe = Range("ringe") ' ringide arv  
  algaeg = Timer()  
  For ring = 1 To ringe ' For-lause algus  
    Range("ring") = ring ' ringi number  
    Do Until auto.Left > 1000  
      Range("aeg") = Timer() - algaeg  
      auto.IncrementLeft 20 + Rnd() * 10  
      paus 0.02  
    Loop  
    auto.Left = 0  
  Next ring ' For –lause lõpp  
End Sub
```

Programm imiteerib auto ringliiklust.

Kasutusel on objekt **auto** kolm lahtrit:

ringe, ring ja **aeg**

neli muutujat:

auto, ringe, ring, algaeg



Autosõit

Animatsiooni loomine VBA abil

VB käsk **DoEvents** võimaldab operatsioonisüsteemil täita ootel olevad tegevused, sh näidata objektide omaduste muutumist töölehel

- ▶ Pärast objektide omaduste (asukoht, suurus, värv jm) muudatust tuleb kasutada käsku DoEvents.
- ▶ Mitme objekti samaaegsete tegevuste määramiseks tuleb vastavad laused panna ühte või mitmesse kordusesse.
- ▶ Tegevuste kirjeldamiseks ja kontrollimiseks võib kasutada eraldi protseduure, millest igaüks määrab ühe sammu.
- ▶ Kui tegevused on mitmes sõltumatus protseduuris, võiks nimega konstandid ja muutujad olla globaalsed.

Alamprotseduurid

Osa programmist võib vormistada (alam)protseduurina

- ▶ kui sama tegevust saab kasutada mitmest kohast (kas ühes või mitmes programmis), võimalusel ka erinevate argumentidega
- ▶ kui see täidab mingit kindlalt piiritletud ülesannet, mis põhiprotseduuris võtab liiga palju ruumi

Alamprotseduurid

- ▶ sub-protseduur
- ▶ funktsioon, kui eesmärgiks on leida üks väärtus

NB! VBAs omistatakse tagastatav väärtus funktsiooni nimele

Parameetrid – muutujad, mis saavad väärtuse peaprotseduurist (väljastpoolt) ja/või tagastavad tulemuse peaprotseduurile.