Základní pojmy algoritmizace a programování

ALGORITMUS
INSTRUKCE
STROJOVÝ KÓD
VYŠŠÍ PROGRAMOVACÍ JAZYK
ZDROJOVÝ KÓD
KOMPILACE

Algoritmus

- Předpis, jak řešit určitý problém (přesně určená konečná posloupnost pravidel, jejichž realizace nám umožní pro přípustné hodnoty vstupních dat nalézt po konečném počtu kroků správná výstupní data). Navrhování a vymýšlení algoritmů je základem programování.

  Podmínky:

  Problému je řešitelný v konečném počtu kroků

  Pravidla lze opakovat libovolněkrát v dané posloupnosti

  Za stejných podmínek musí dát vždy stejné výsledky

Příklad algoritmu

- Algoritmus pro určení velikosti přepony pravoúhlého trojúhelníku:
  - Vezmi kvadrát velikosti jedné odvěsny, zvětši o kvadrát velikosti druhé odvěsny, výsledek odmocni; dostaneš velikost přepony
- Algoritmus pro sestavení skříňky nakoupené v obchodním domě IKEA
  - Obrázkový návod k sestavení v jednotlivých krocích

Instrukce (strojová – počítače)

- Elementární operační krok počítače.
  - Př: zvětšení číselné hodnoty o 1; porovnání dvou číselných hodnot podle velikosti; nalezení konkrétního znaku v textu

Za jednu sekundu zpracuje procesor několik miliard mikroinstrukcí

V určitém typu procesoru počítače jsou pomocí logických obvodů vytvořeny instrukce v tzv. instrukční sadě

Příklad instrukce pro procesor řady i386:

F0 6626 67C7 845E 7856 3412 F0DE BC9A - lock mov dword [es:esi+ebx*2+0x12345678],0x9abcdef0.

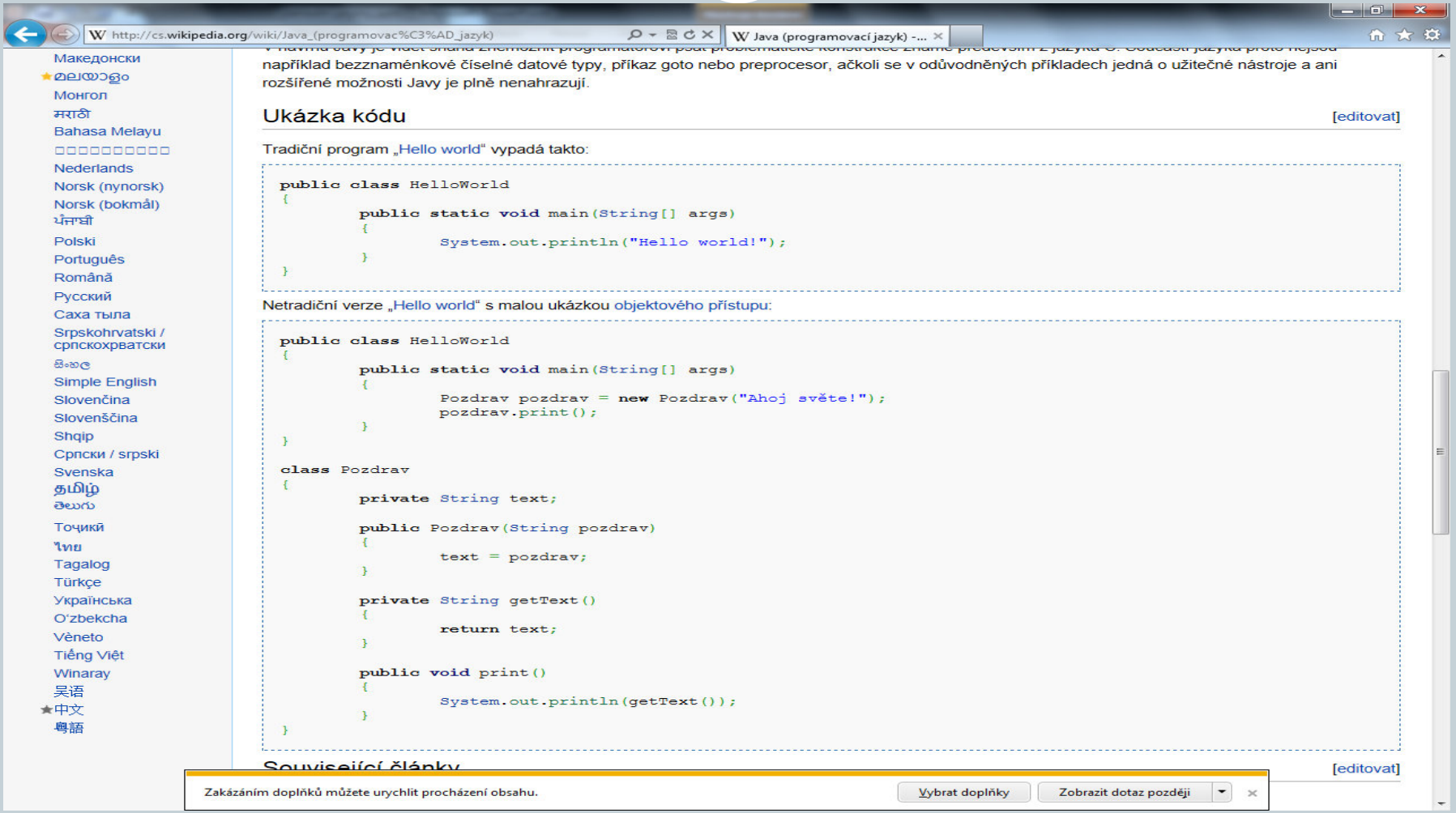
Program (strojový – počítačový)

- Posloupnost instrukcí, jejichž vykonání realizuje algoritmus
  - Programování přímo ve strojovém kódu je značně pracné,
  - Strojový kód vzniká většinou překladem zdrojového kódu psaného ve vyšších programovacích jazycích do strojového kódu příslušného procesoru.
  - Překladu do strojového kódu se obecně vzato nelze vyhnout, každá aplikace nějakým způsobem musí běžet ve strojovém kódu, protože procesor žádný jiný kód z principu své funkce nemůže přímo interpretovat.

Programovací jazyk vyšší

- Souhrn syntaktických pravidel pro zápis algoritmu ve formě převoditelné na posloupnost instrukcí vykonatelných počítačem (strojových) nebo ve formě interpretovatelné posloupnosti vykonatelných instrukcí.
  - Př: QBasic, Visual Basic, C#, C++, COBOL, Pascal, Jawa Small Basic (zdarma ke stažení na stránkách Microsoft)

Ukázka kódu - JAVA



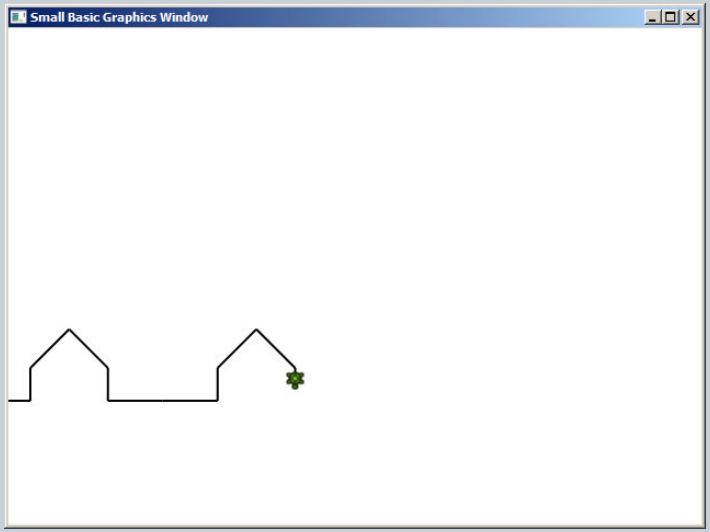
Program

- Konkrétní zápis (jde vždy o text) konkrétního algoritmu podléhající syntaktickým pravidlům daného programovacího jazyka.

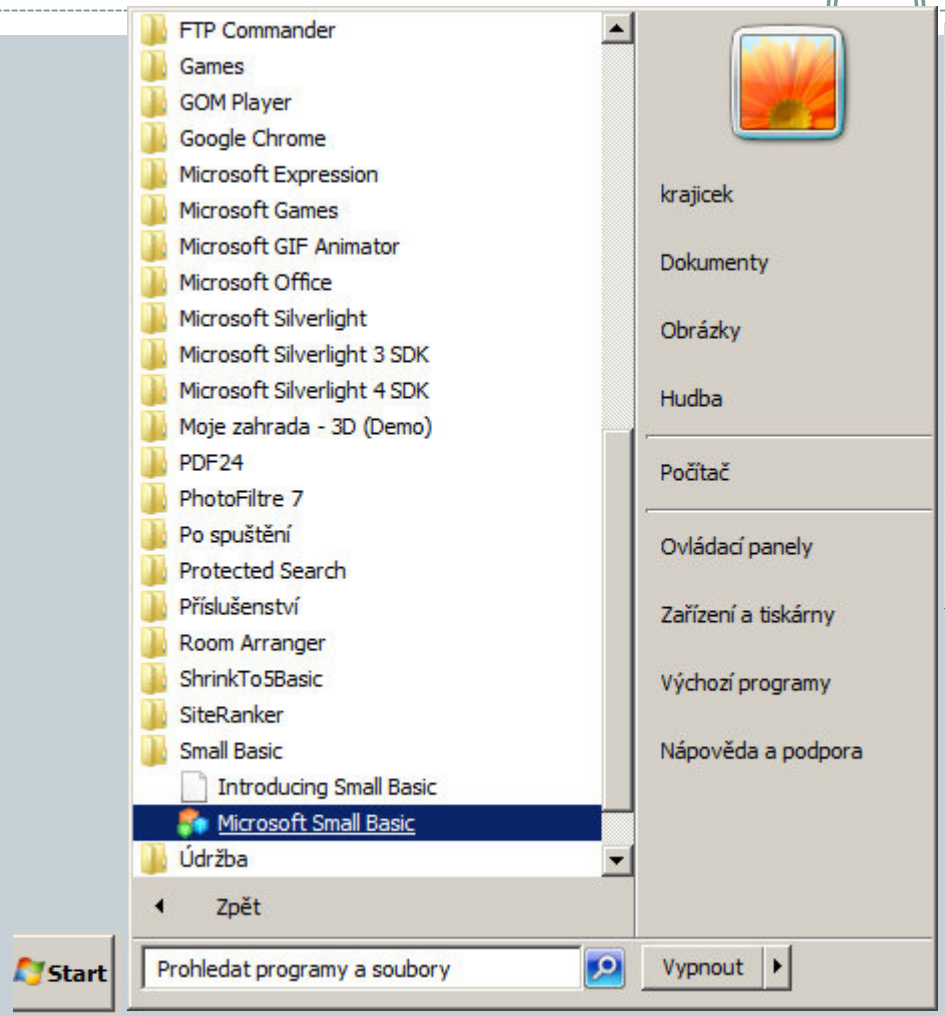
- Překlad programu (kompilace)
  - Převod textu programu vyššího programovacího jazyka do strojového kódu.

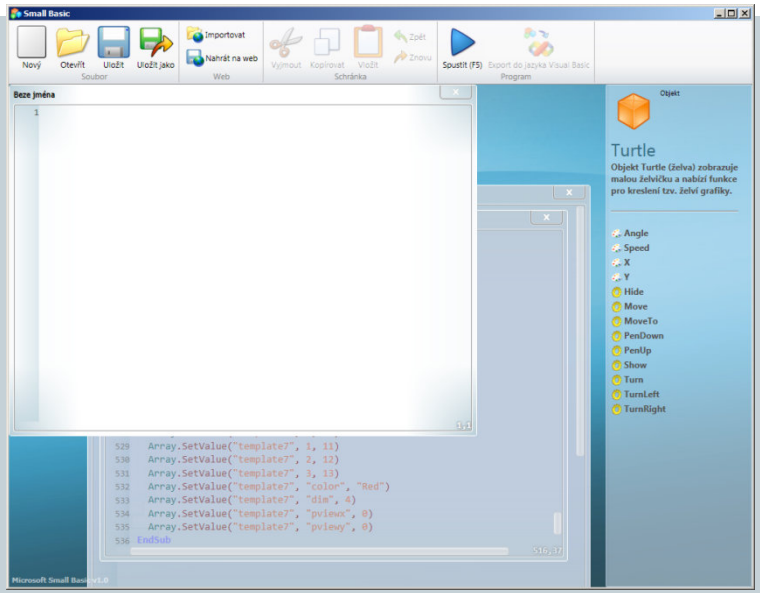
Ukázka programového kódu Small Basic

```
Turtle.Show()
Turtle.Speed= 5
Turtle.penup ()
Turtle.Move (-100)
Turtle.Turn (-90)
Turtle.Move (350)
Turtle.turn (90)
Turtle.pendown ()
Turtle.turn (90)
For j=1 To 4
  Turtle.Move (50)
  Turtle.Turn (-90)
  Turtle.Move (30)
  Turtle.Turn(45)
  turtle.move (50)
  Turtle.Turn (90)
  turtle.move (50)
  Turtle.Turn(45)
  Turtle.Move (30)
  Turtle.Turn (-90)
  Turtle.Move (50)
endfor
```



Spuštění prostředí vyššího programovacího jazyka Small Basic





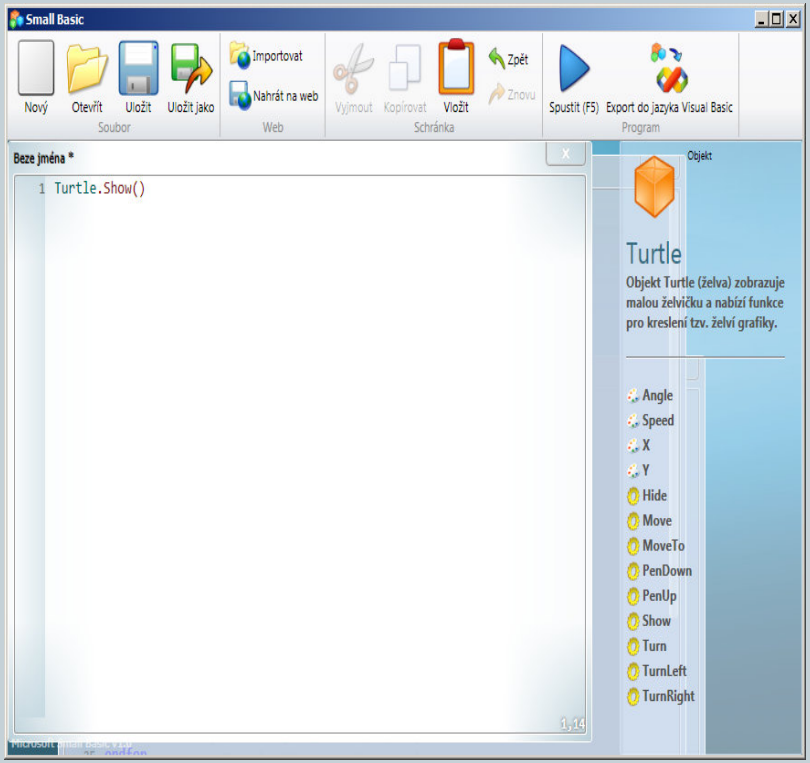
Příklad vytvořeného programu:

Tetris

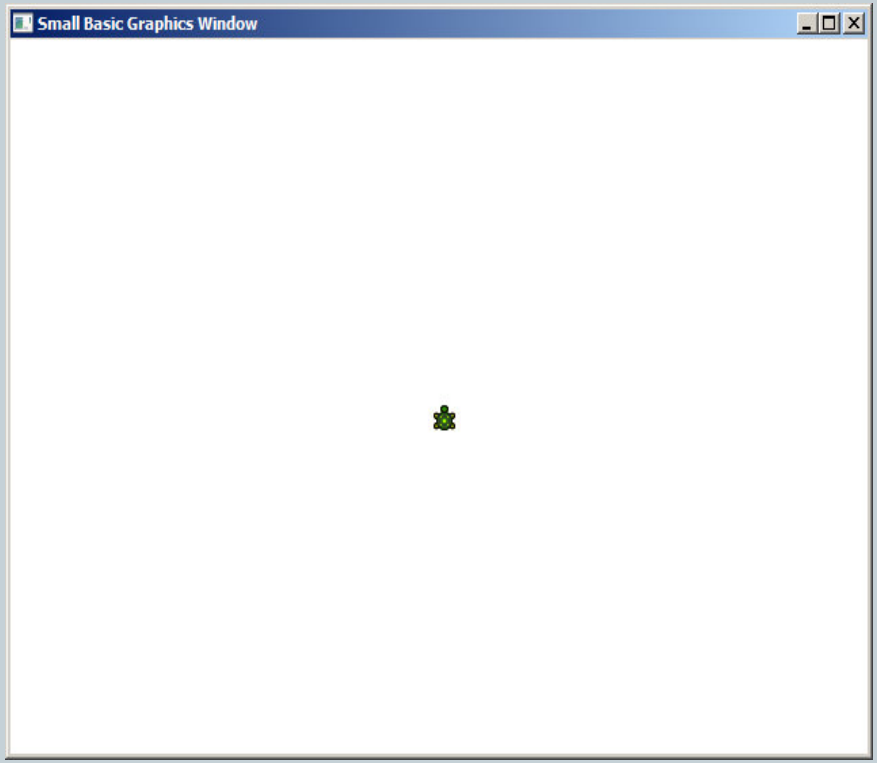
Zdrojový kód: tetris

1. Program v Small Basic – želvička kreslí

Spusťte program Small Basic a zapište kód: Turtle.Show()

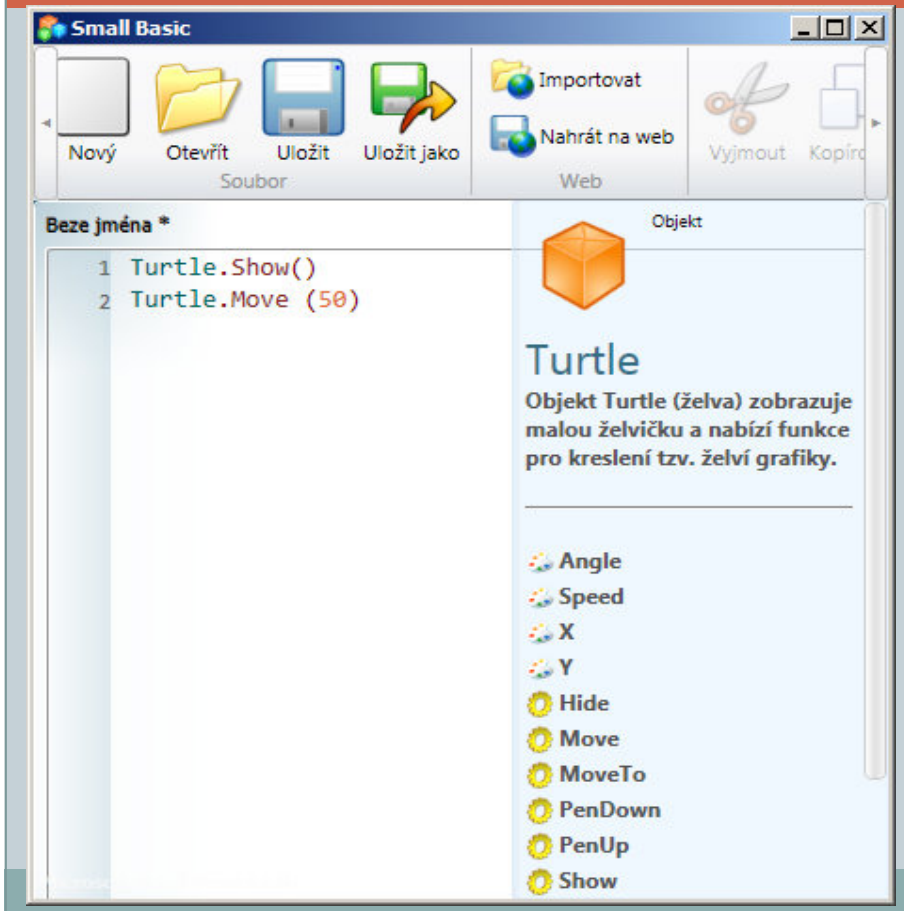


Zvolte spustit nebo F5.

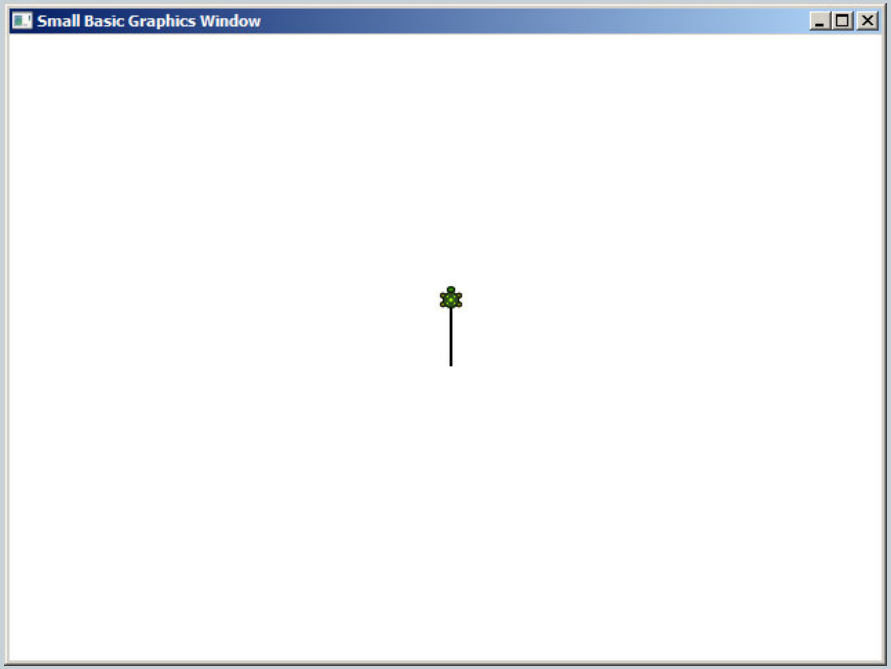


Želva se pohybuje a kreslí čáru

Zapište:Turtle.Move (50)



Výsledek



- Zapište kód:

Pro otočení o 90° vlevo zapište:

```
Turtle.Turn (-90)
```

Pro pohyb o 50 bodů:

```
Turtle.Move (50)
```

Pro otočení o 90° vpravo a pohyb

```
Turtle.turn (90)
Turtle.Move (50)
```

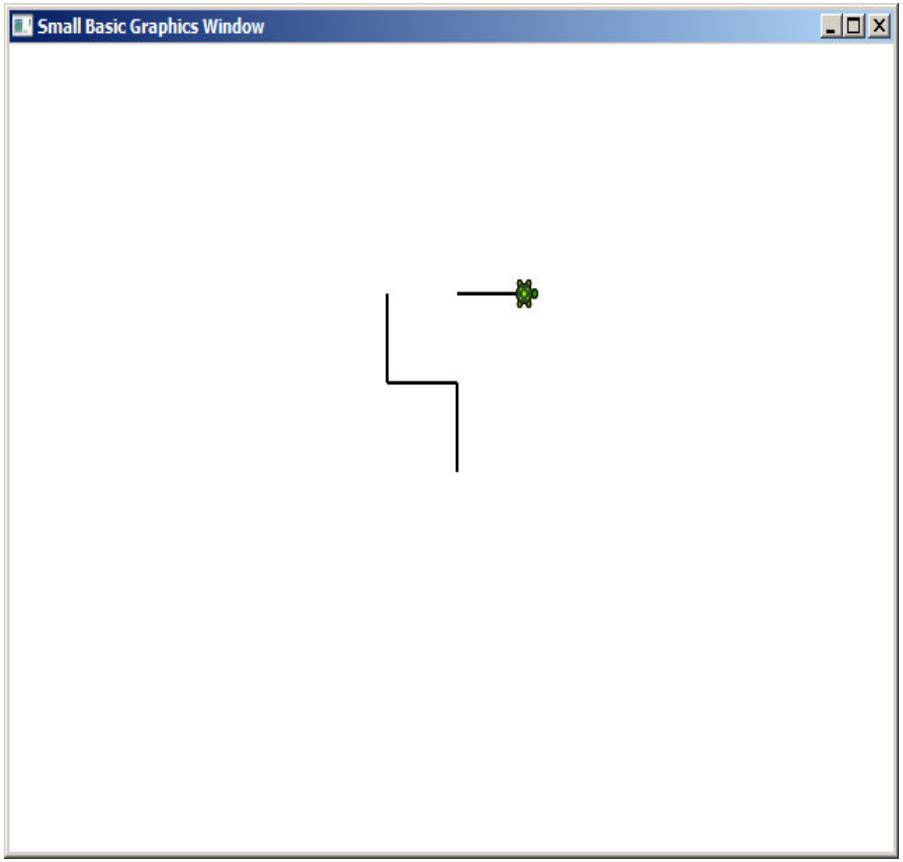
Pro zvednutí prdelky a pohyb:

```
Turtle.penup ()
Turtle.turn (90)
Turtle.Move (50)
```

Pro začátek kreslení a pohyb:

```
Turtle.pendown ()
Turtle.Move (50)
```

- Výsledek:

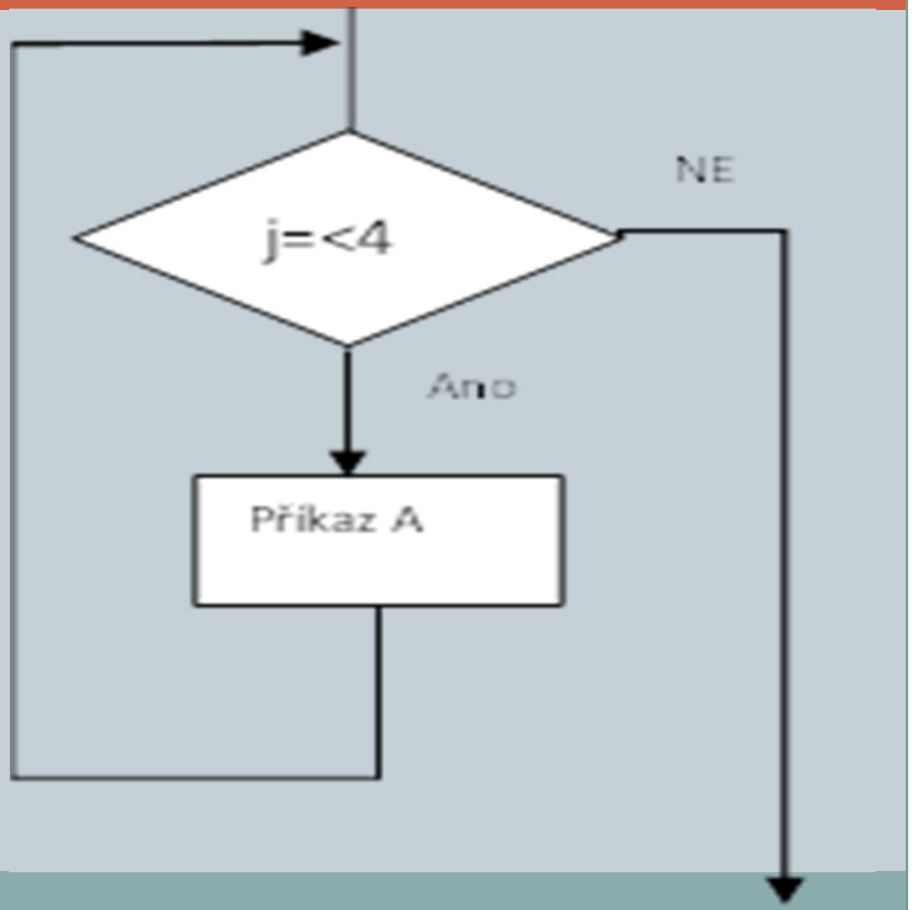


2. Program v Small Basic – želvička kreslí domeček

Programová struktura cyklus o předem známém počtu opakování

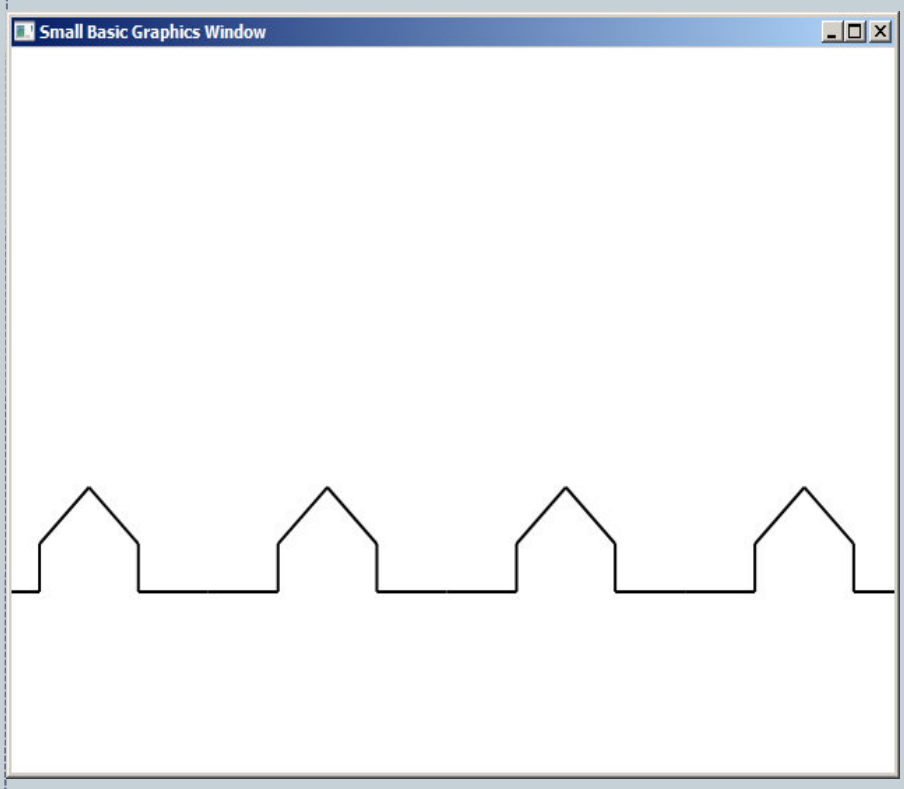
- **For j=1 To 4**
- Turtle.Move (50)
- Turtle.Turn (-90)
- Turtle.Move (30)
- Turtle.Turn(45)
- turtle.move (50)
- Turtle.Turn (90)
- turtle.move (50)
- Turtle.Turn(45)
- Turtle.Move (30)
- Turtle.Turn (-90)
- Turtle.Move (50)
- **endfor**

Vývojový diagram



Celý program

```
Turtle.Show()
Turtle.Speed= 5
Turtle.penup ()
Turtle.Move (-100)
Turtle.Turn (-90)
Turtle.Move (350)
Turtle.turn (90)
Turtle.pendown ()
Turtle.turn (90)
For j=1 To 4
  Turtle.Move (50)
  Turtle.Turn (-90)
  Turtle.Move (30)
  Turtle.Turn(45)
  turtle.move (50)
  Turtle.Turn (90)
  turtle.move (50)
  Turtle.Turn(45)
  Turtle.Move (30)
  Turtle.Turn (-90)
  Turtle.Move (50)
endfor
```



```
GraphicsWindow.KeyDown = HandleKey
GraphicsWindow.BackgroundColor = GraphicsWindow.GetColorFromRGB( 253, 252, 251 )

While "True"
  BOXES = 4      ' number of boxes per piece
  BWIDTH = 25    ' box width in pixels
  XOFFSET = 40   ' Screen X offset in pixels of where the board starts
  YOFFSET = 40   ' Screen Y offset in pixels of where the board starts
  CWIDTH = 10    ' Canvas Width, in number of boxes
  CHEIGHT = 20   ' Canvas Height, in number of boxes.
  STARTDELAY = 800
  ENDDELAY = 175
  PREVIEW_xpos = 13
  PREVIEW_ypos = 2

  GraphicsWindow.Clear()
  GraphicsWindow.Title = "Small Basic Tetris"
  GraphicsWindow.Height = 580
  GraphicsWindow.Width = 700
  GraphicsWindow.Show()

  SetupTemplates()
  SetupCanvas()
  MainLoop()

  GraphicsWindow.ShowMessage( "Game Over", "Small Basic Tetris" )
EndWhile

Sub MainLoop
  template = Text.Append("template", Math.GetRandomNumber(7))

  CreatePiece() ' in: template ret: h
  nextPiece = h

  end = 0
  sessionDelay = STARTDELAY
  While end = 0
    If sessionDelay > ENDDELAY Then
      sessionDelay = sessionDelay - 1
    EndIf

    delay = sessionDelay
    thisPiece = nextPiece
    template = Text.Append("template", Math.GetRandomNumber(7))

    CreatePiece() ' in: template ret: h
    nextPiece = h
    DrawPreviewPiece()

    h = thisPiece

    ypos = 0
    done = 0
```

zpět

Další příkazy

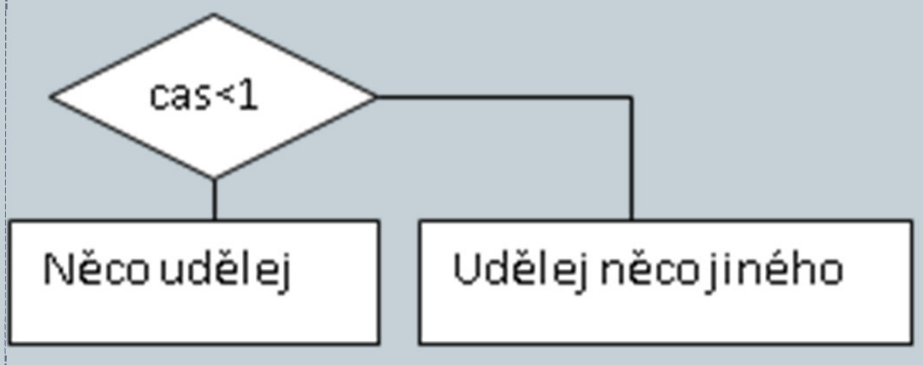
- Otevření textového okna, volba barvy písma, zobrazení textu
  - TextWindow.ForegroundColor = "Yellow"
  - TextWindow.WriteLine("Ahoj světe!")
- Zadání hodnoty proměnné, zobrazení obsahu proměnné
  - TextWindow.Write("Napište své jméno: ")
  - jmeno = TextWindow.Read()
  - TextWindow.WriteLine("Ahoj " + jmeno)

Další příkazy

- Součet hodnot proměnných
  - cislo1 = 10
  - cislo2 = 20
  - cislo3 = cislo1 + cislo2
  - TextWindow.WriteLine(cislo3)
- Součet hodnot proměnných se zadáním hodnot
  - TextWindow.Write("Zadejte první číslo ")
  - Cislo1 = TextWindow.ReadNumber()
  - TextWindow.Write("Zadejte druhé číslo ")
  - Cislo2 = TextWindow.ReadNumber()
  - vysledek = cislo1 + cislo2
  - TextWindow.WriteLine(" součet čísel je " + vysledek)

Programová struktura větvení

- Načtení systémového času, podmínka když
  - If (Clock.Hour < 12) Then
  - TextWindow.WriteLine(" Dobré ráno, světe")
  - EndIf
  - If (Clock.Hour >= 12) Then
  - TextWindow.WriteLine(" Dobrý večer, světe")
  - EndIf



Příkaz skoku ve struktuře větvení (opakování)

```
i=1
start:
TextWindow.WriteLine(i)
i = i + 1
If (i < 25) Then
Goto start
EndIf
```

- Pokuste se vytvořit stejný program cyklem s předem známým počtem opakování
  FOR

Cyklus s předem známým počtem opakování

```
For i = 1 To 24
TextWindow.WriteLine(i)
EndFor
```

Vlastnost step (krok)

- For i = 10 To 1 Step -1
- TextWindow.WriteLine(i)
- EndFor

Cyklus dokud podmínka platí

- Příkaz While (zatímco, dokud)
  - cislo = 100
  - While (cislo > 1)
  - TextWindow.WriteLine(cislo)
  - cislo = cislo / 2
  - EndWhile

Vlastnosti kreslené čáry

- Želva kreslí červenou čarou
  - GraphicsWindow.PenColor = "red„
  - Turtle.Show()
  - Turtle.Move (50)
- A čárou tlustou 10 bodů
  - GraphicsWindow.PenWidth = 10

Kreslení tvarů

- Čára
  - GraphicsWindow.DrawLine(10, 100, 100, 10)
- Obdélník prázdný
  - GraphicsWindow.DrawRectangle(10, 13, 300, 60)
- Obdélník plný
  - GraphicsWindow.FillRectangle(10, 200, 300, 60)
- Elipsa
  - GraphicsWindow.DrawEllipse(10, 320, 300, 60)

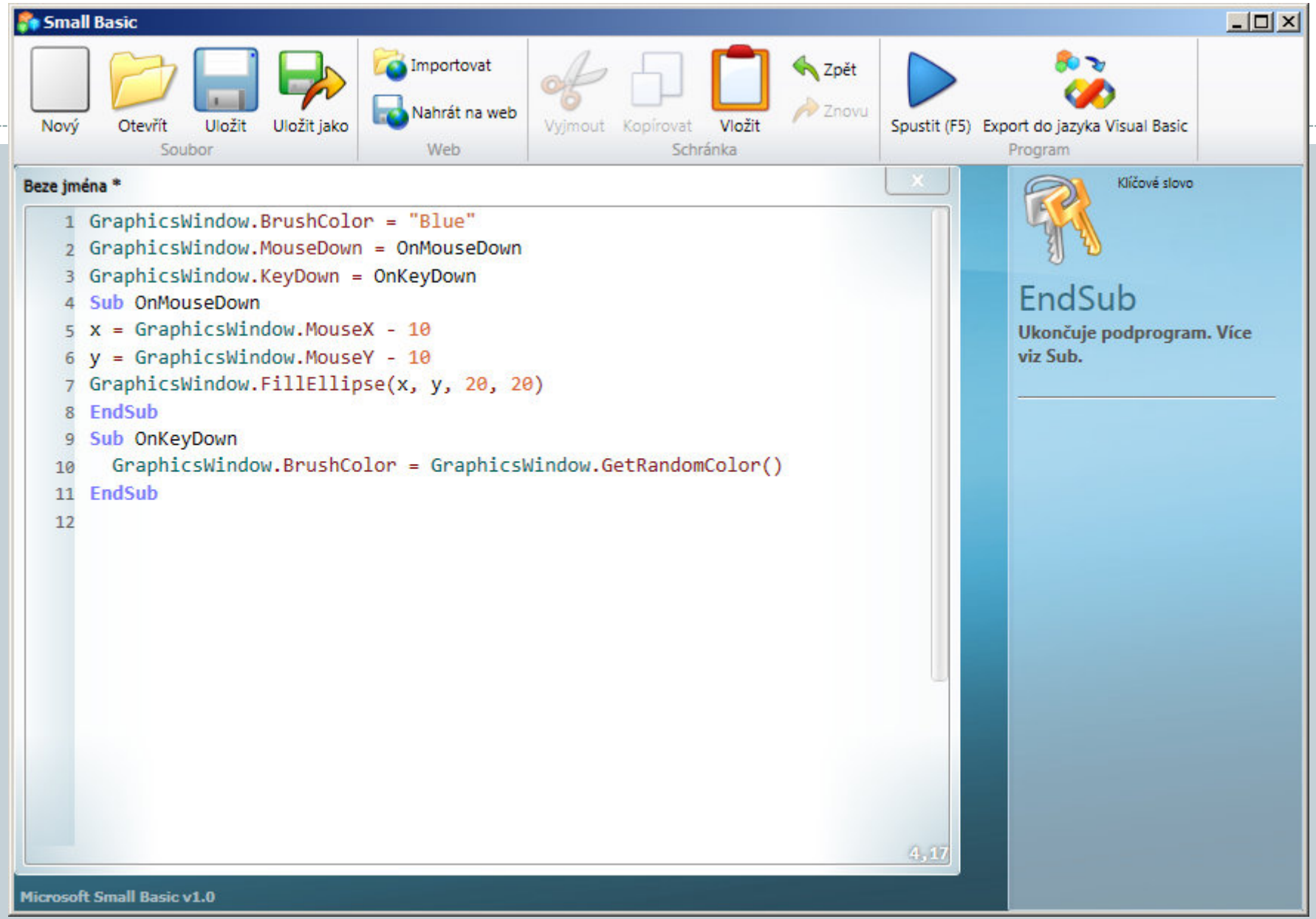
- Barvy výplně
  - GraphicsWindow.BrushColor = "Green"
- Barva čáry
  - GraphicsWindow.PenColor = "Red"

Soustředné kružnice

- GraphicsWindow.BackgroundColor = "Black"
- GraphicsWindow.PenColor = "LightGreen"
- GraphicsWindow.Width = 200
- GraphicsWindow.Height = 200
- For i = 1 To 100 Step 5
- GraphicsWindow.DrawEllipse(100 - i, 100 - i, i * 2, i * 2)
- EndFor

Použití náhody

- Náhodná barva
  - GraphicsWindow.PenColor = GraphicsWindow.GetRandomColor()
- Pole, animace, zpoždění běhu programu
  - radky = 8
  - sloupce = 8
  - rozmer = 40
  - **For r = 1 To radky**
  - **For s = 1 To sloupce**
  - GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()
  - ctverce[r][s] = Shapes.AddRectangle(rozmer, rozmer)
  - Shapes.Move(ctverce[r][s], s * rozmer, r * rozmer)
  - **EndFor**
  - **EndFor**
  - **For r = 1 To radky**
  - **For s = 1 To sloupce**
  - Shapes.Animate(ctverce[r][s], 0, 0, 1000)
  - Program.Delay(300)
  - **EndFor**
  - **EndFor**


Small Basic
Nový
Otevřít
Uložit
Uložit jako
Soubor
Importovat
Nahrát na web
Web
Vyjmout
Kopírovat
Vložit
Zpět
Znovu
Schránka
Spustit (F5)
Export do jazyka Visual Basic
Program
Beze jména *
1 GraphicsWindow.BrushColor = "Blue"
2 GraphicsWindow.MouseDown = OnMouseDown
3 GraphicsWindow.KeyDown = OnKeyDown
4 Sub OnMouseDown
5 x = GraphicsWindow.MouseX - 10
6 y = GraphicsWindow.MouseY - 10
7 GraphicsWindow.FillEllipse(x, y, 20, 20)
8 EndSub
9 Sub OnKeyDown
10 GraphicsWindow.BrushColor = GraphicsWindow.GetRandomColor()
11 EndSub
12
4, 17
Klíčové slovo
EndSub
Ukončuje podprogram. Více viz Sub.
Microsoft Small Basic v1.0

Ovládání pohybem myši

```
plosina = Shapes.AddRectangle(120, 12)
GraphicsWindow.MouseMove = OnMouseMove
Sub OnMouseMove
plosinaX = GraphicsWindow.MouseX
Shapes.Move(plosina, plosinaX - 60,
GraphicsWindow.Height - 12)
EndSub
```

Pohyb míčku v uzavřeném prostoru

- micek = Shapes.AddEllipse(16, 16)
- gw = GraphicsWindow.Width
- gh = GraphicsWindow.Height
- x = 0
- y = 0
- deltaX = 1
- deltaY = 1
- Nahoru:
- x = x + deltaX
- y = y + deltaY
- **If (x >= gw or x <= 0) Then**
- deltaX = -deltaX
- **EndIf**
- **If (y <= 0 Or y>=gh) Then**
- deltaY = -deltaY
- **EndIf**
- Shapes.Move(micek, x, y)
- Program.Delay(5)
- **Goto nahoru**

Hra ARKANOID

```
GraphicsWindow.BackgroundColor = "DarkBlue"
plosina = Shapes.AddRectangle(120, 12)
micek = Shapes.AddEllipse(16, 16)
GraphicsWindow.MouseMove = OnMouseMove
x = 0
y = 0
deltaX = 1
deltaY = 1
Nahoru:
x = x + deltaX
y = y + deltaY
gw = GraphicsWindow.Width
gh = GraphicsWindow.Height
If (x >= gw - 16 or x <= 0) Then
  deltaX = -deltaX
EndIf
If (y <= 0) Then
  deltaY = -deltaY
EndIf
padX = Shapes.GetLeft (plosina)
If (y = gh - 28 and x >= padX and x <= padX + 120) Then
  deltaY = -deltaY
EndIf
Shapes.Move(micek, x, y)
Program.Delay(5)
If (y < gh) Then
  Goto Nahoru
EndIf
GraphicsWindow.ShowMessage("Prohráli jste", "Arkanoid")
Sub OnMouseMove
  plosinaX = GraphicsWindow.MouseX
  Shapes.Move(plosina, plosinaX - 60, GraphicsWindow.Height - 12)
EndSub
```