
Programavimo kalbos mokykloje

Albertas Dinda

2009

Logo šeima

- Logo atsirado maždaug 1967 m.
- Logo Writer (DOS, 1995-1997 m.)
- Komenskio Logo (Win, 1998 m.)
- Imagine Logo (Win, 2006 m.)
- Elica (Win, 2007m.)
- Projektas Dalest
- Scratsh
- Dabar yra apie 200 šios kalbos dialektų
- 100% efektyvumo kalba

Komenskio Logo

- Pamokos
 - <http://aldona.mii.lt/pms/Logo/tekstai/tpamokos.html>
- Nupirhta visoms Lietuvos mokykloms, laisvai platinama

Imagine Logo (Slovakija)

- Lokalizuota, galima įsigyti, jau buvo platinama Lietuvoje
- Multimedia (interaktyvios pateiktys)
- Darbas su garsais (muzikinis redaktorius)
- Kompiluoja .exe bylas
- Objektinio programavimo lygis
- Svetainė <http://www.logo.lt/imagine.htm>

Elica (Bulgarija)

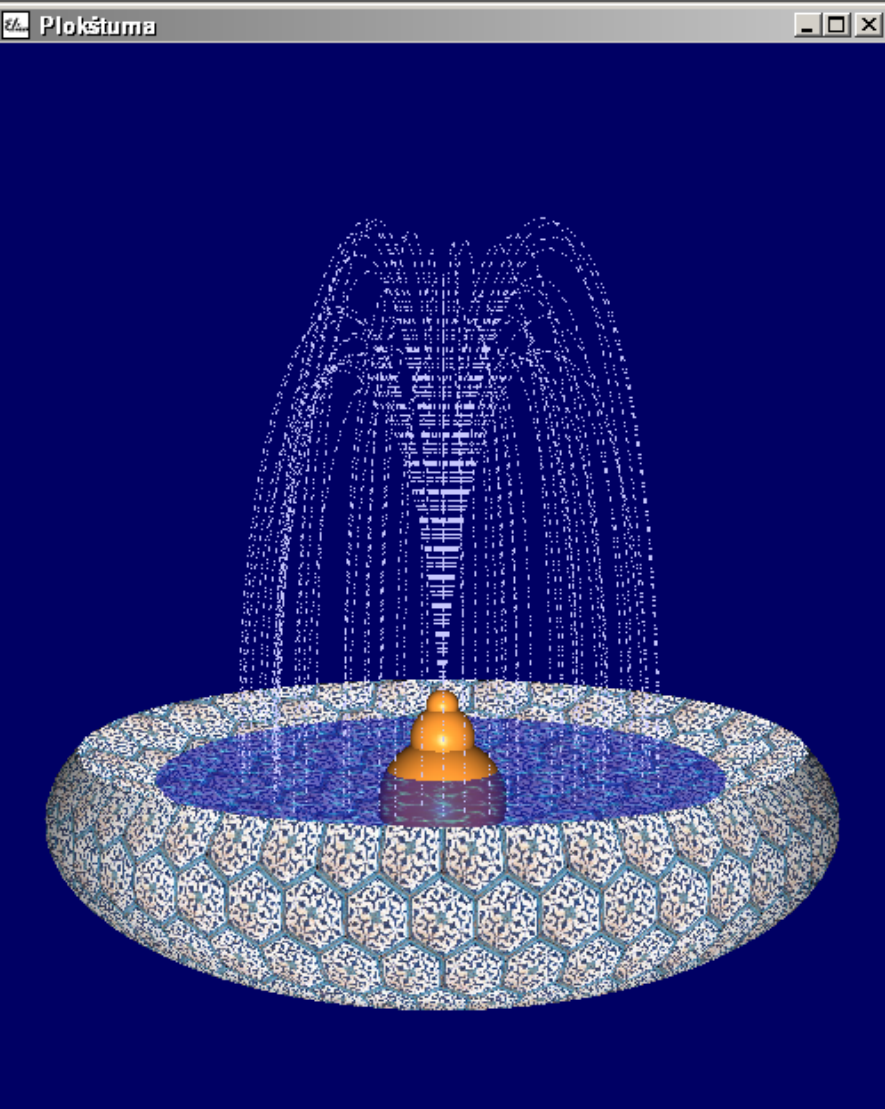
- 3D Logo sistema
- Objektinis lygis
- Virtuali realybė
- Nedidelė apimtis (5MB)
- Tinklapis
 - <http://www.elica.net/site/about/about.html>
- Lietuviška sąsaja (lengvai keičiama)
- Naudojama JAV, Japonijoje, Europoje ...
- Laisvai platinamas
- Virš 600 pavyzdžių biblioteka.

Aplinka ir programos pavyzdys „Fontanas“

Elica 5.6 (build 15)

Failas Taisa Rodymas Vykdymas Nuostatos Žinynas | 11.837 s

Plokštuma



Water Fountain.eli

```
run "graphix
screen [500x600 [rgb 0 0 100]]

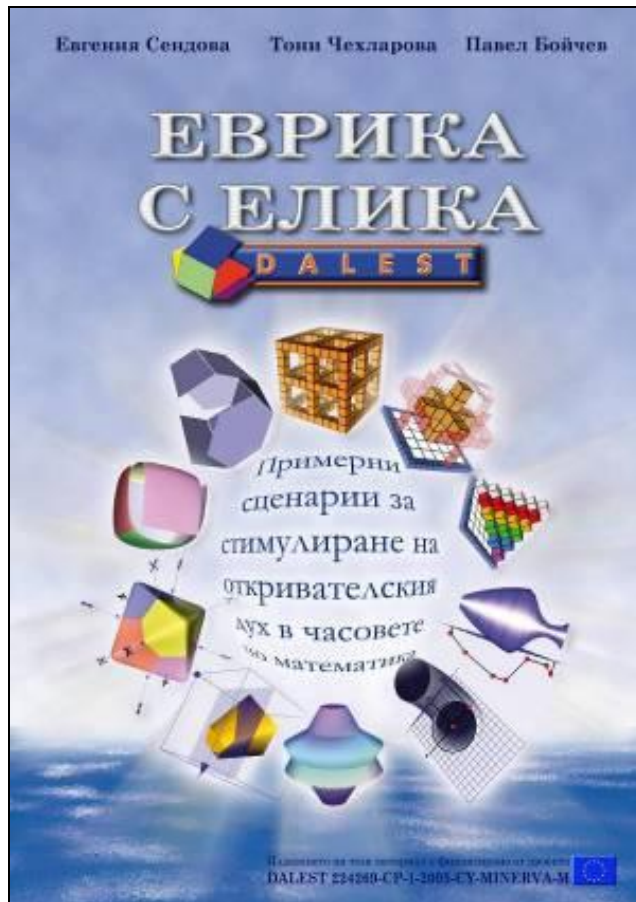
to POOL
  make local "mode 2
  make local "color rgb 255 255 255
  make local "smooth "true
  make local "precision 64
  make local "top custom sphere vector 0 0 -2 12 (set
    "light "true
    "texture texture "Tile2.jpg
    "texture.scale 1
    "texture.smooth "true
    "radiusz 5
  )
  make local "bottom custom sphere vector 0 0 22.4 25 (set
    "texture texture "Tile2.jpg
    "texture.scale 1.5
    "texture.smooth "true
    "normal -1
    "focus vector 1 0 0
  )
)

to ondrawimage
  pushmatrix
  clipplane "true 1 0 0 -1 0
  ondrawimagenow "top
  ondrawimagenow "bottom
  clipplane "false 1
  popmatrix
end
end

make "water custom circle (vector 0 0 -1) 10 (set
  "mode 2
  "texture texture "water006.jpg
  "color (rgb 255 255 255 150)
  "texture.scale 3
```

Dalest projektas (Bulgarija)

Developing and Active Learning Environment for STereometry



- 3D geometrijos objektai
- Sukiniai, pjūviai
- Išklotinės
- Trimatis konstravimas...

Dokumentacija:

<http://www.elica.net/site/papers/papers.html>

<http://www.elica.net/site/museum/Dalest/dalest.html>

Sukiniai

Графика

Без формул | Вкл. бутылката | Точки: 2 3 4 5 6 7

$S = \pi(R+r) \sqrt{(R-r)^2 + h^2}$
 $V = (1/3)\pi(R^2 + Rr + r^2)h$

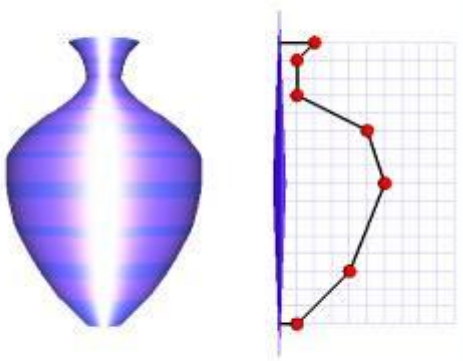
$R_1 = 2, r_1 = 1, h_1 = 1$
 $S_1 = 3\pi \sqrt{3^2 + 1^2} \approx 4.2\pi$
 $V_1 = (1/3)1\pi(2^2 + 2 \cdot 1 + 1^2) = 2.3\pi$

$R_2 = 1, r_2 = 1, h_2 = 2$
 $S_2 = 2\pi \sqrt{0^2 + 2^2} \approx 4\pi$
 $V_2 = (1/3)2\pi(1^2 + 1 + 1^2) = 2\pi$

$R_3 = 1, r_3 = 5, h_3 = 2$
 $S_3 = 6\pi \sqrt{4^2 + 2^2} \approx 26.8\pi$
 $V_3 = (1/3)2\pi(1^2 + 5 + 5^2) = 20.7\pi$

$R_4 = 5, r_4 = 6, h_4 = 3$
 $S_4 = 11\pi \sqrt{1^2 + 3^2} \approx 34.8\pi$
 $V_4 = (1/3)3\pi(5^2 + 30 + 6^2) = 91\pi$

$R_5 = 6, r_5 = 4, h_5 = 5$



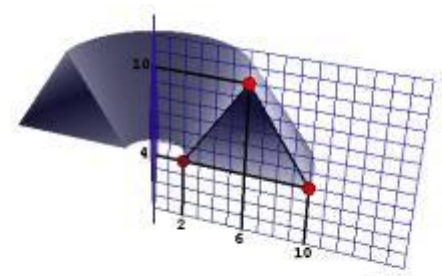
Край

„Butelių dizaina“

Графика

Проекция | Стоби

Съпка 2: Ето обекта след завъртането.




Край

*„Matematiniai
sukiniai“*

Проекция | Стоби

Ето вашето порцеланово изделие.

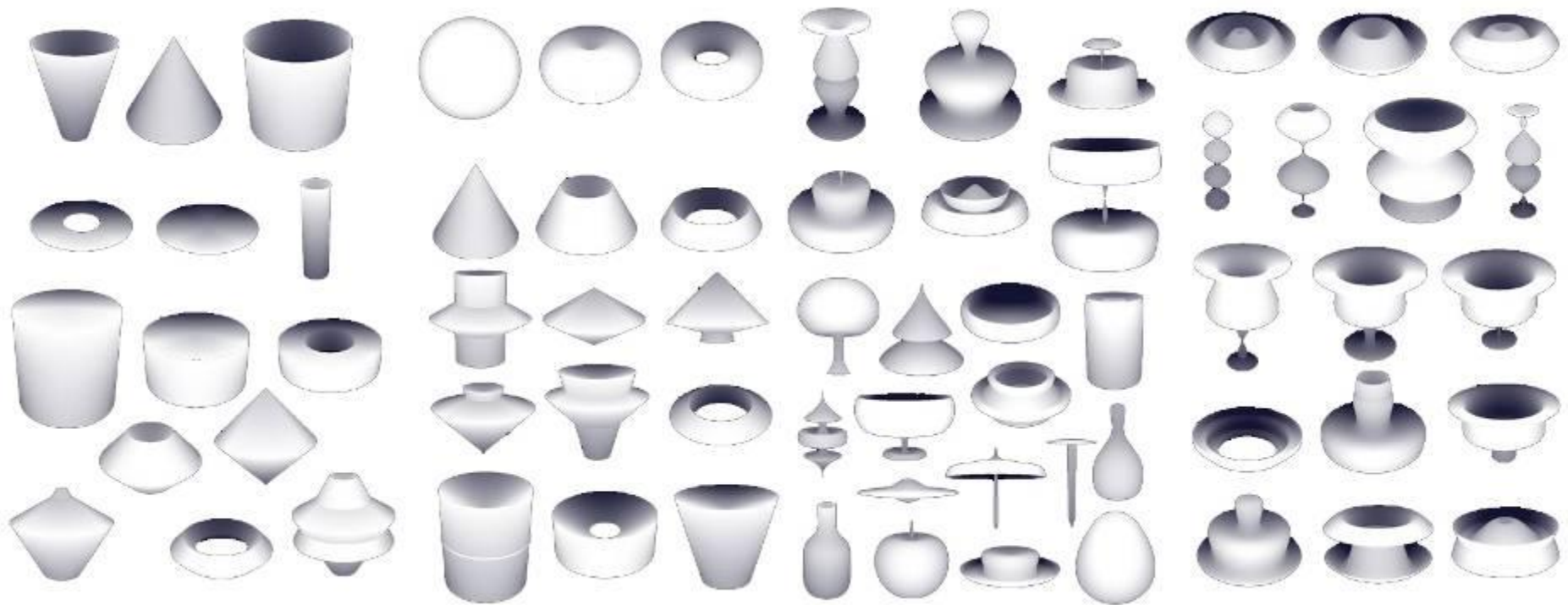


Друга задача | Друга група

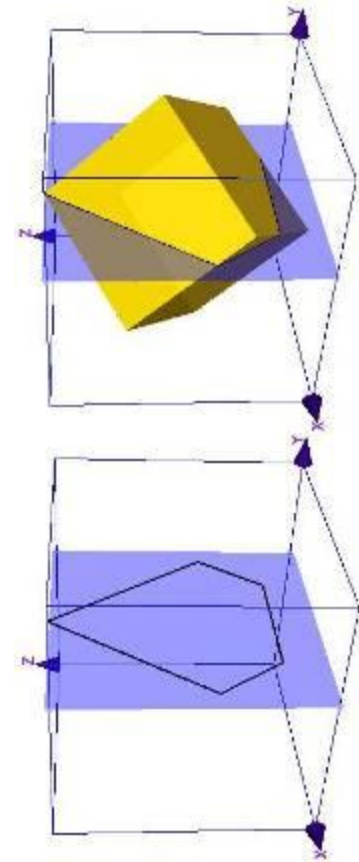
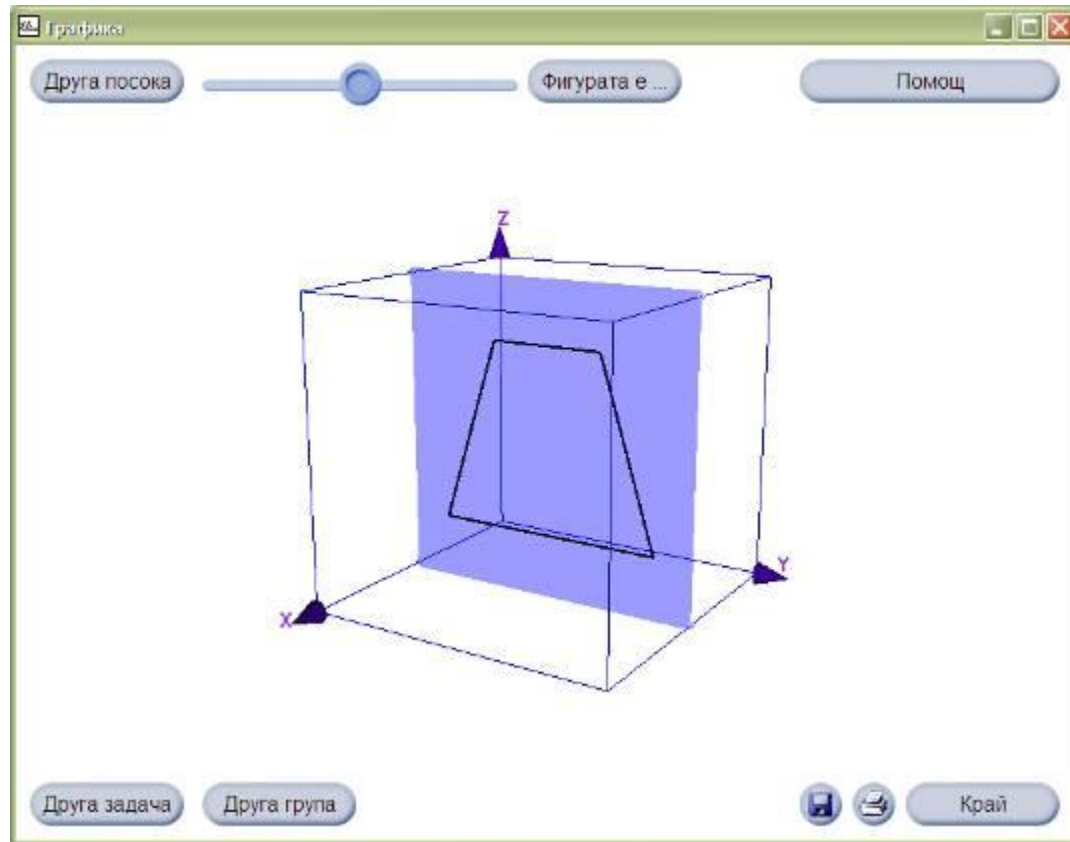
Край

„Žiedimo ratas“

Fantazijos iš molio

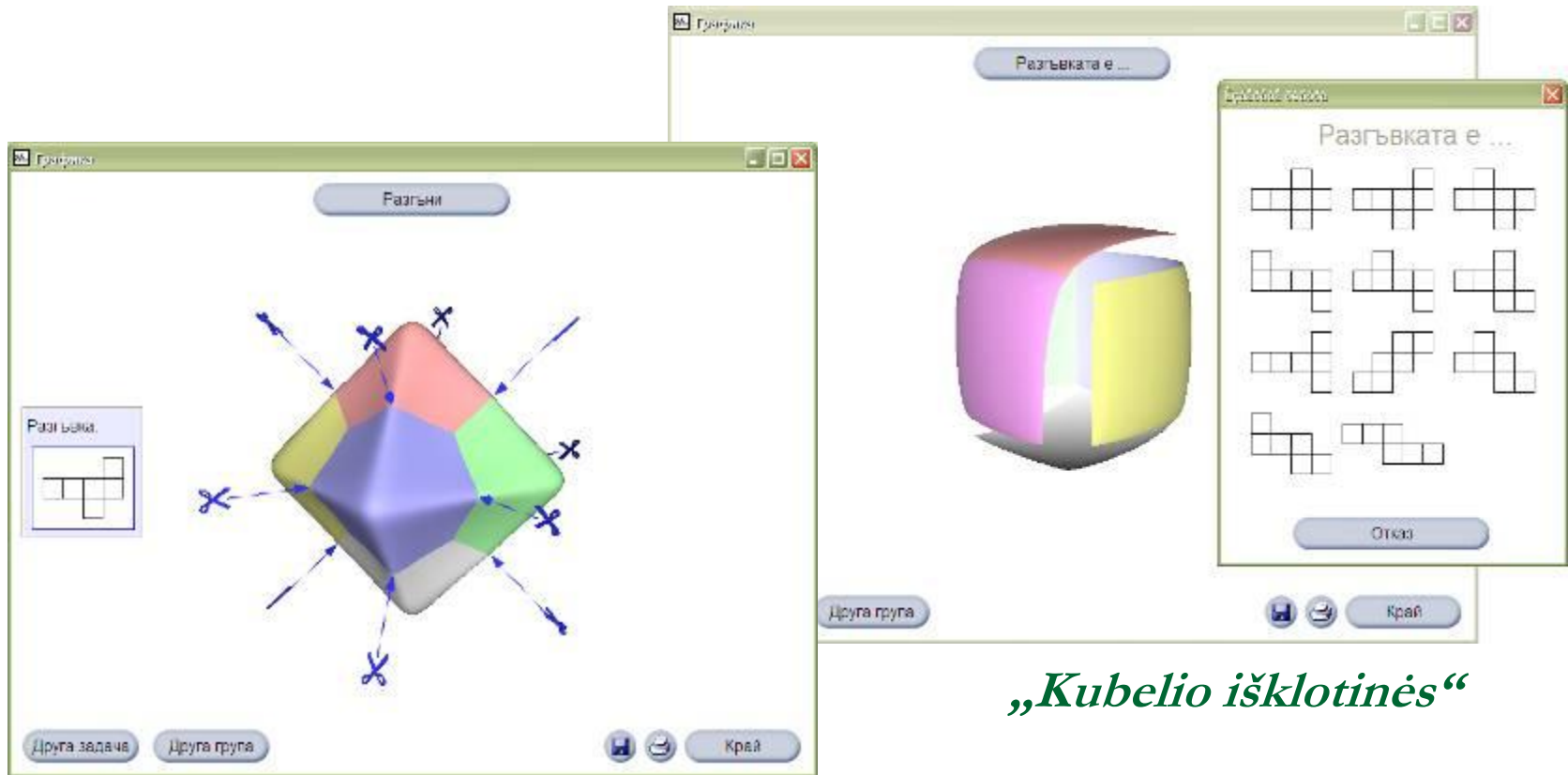


Pjaustyklė



“Pjaustymas”

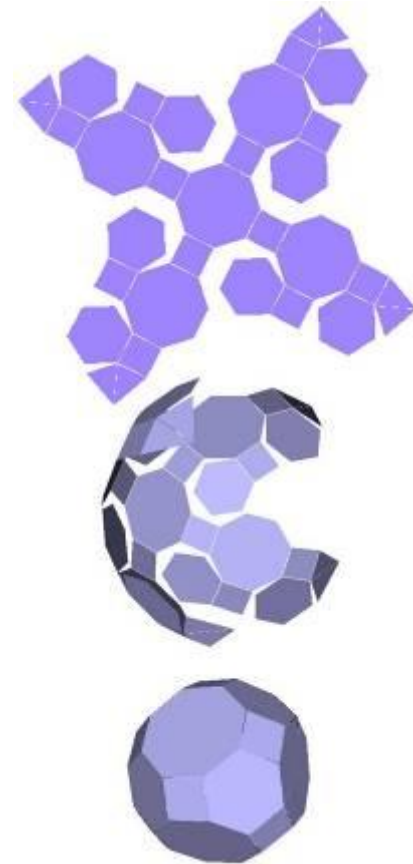
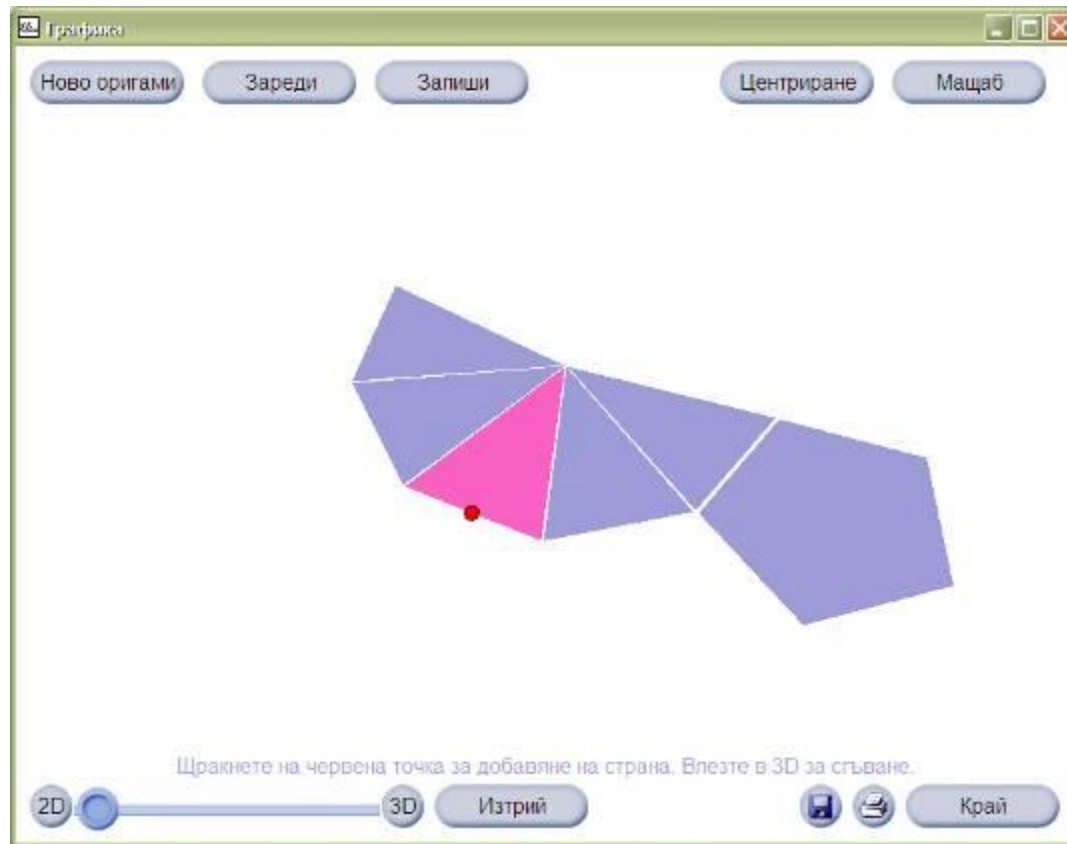
Išklotinės



„Žirkutės“

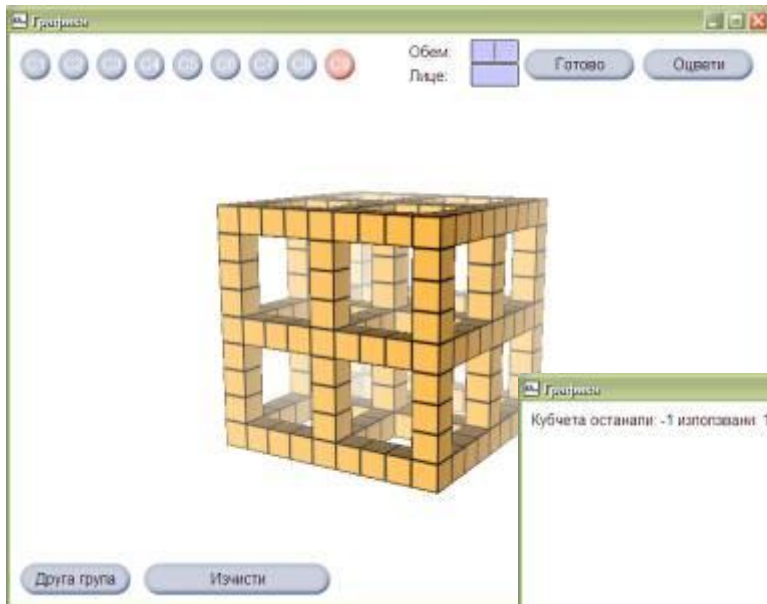
„Kubelio išklotinės“

Origamiai



„Origamiai“

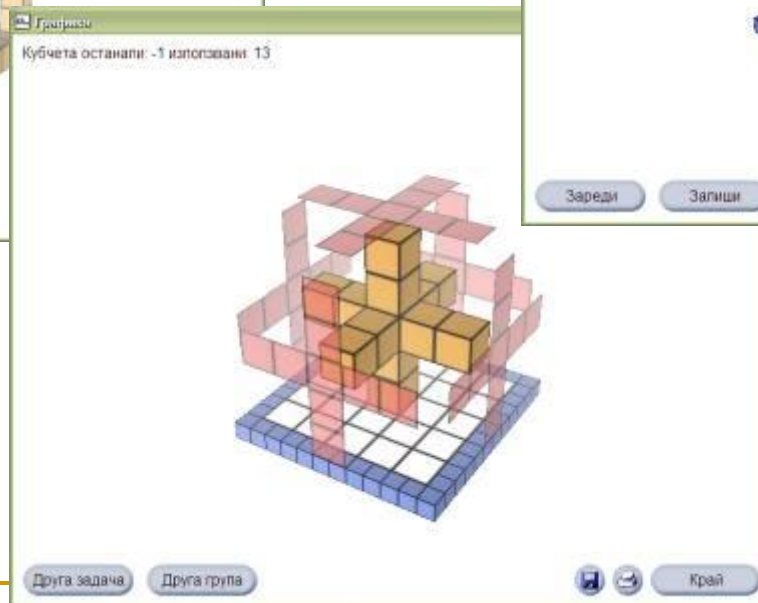
Statybininkas



„Kubelizmas“

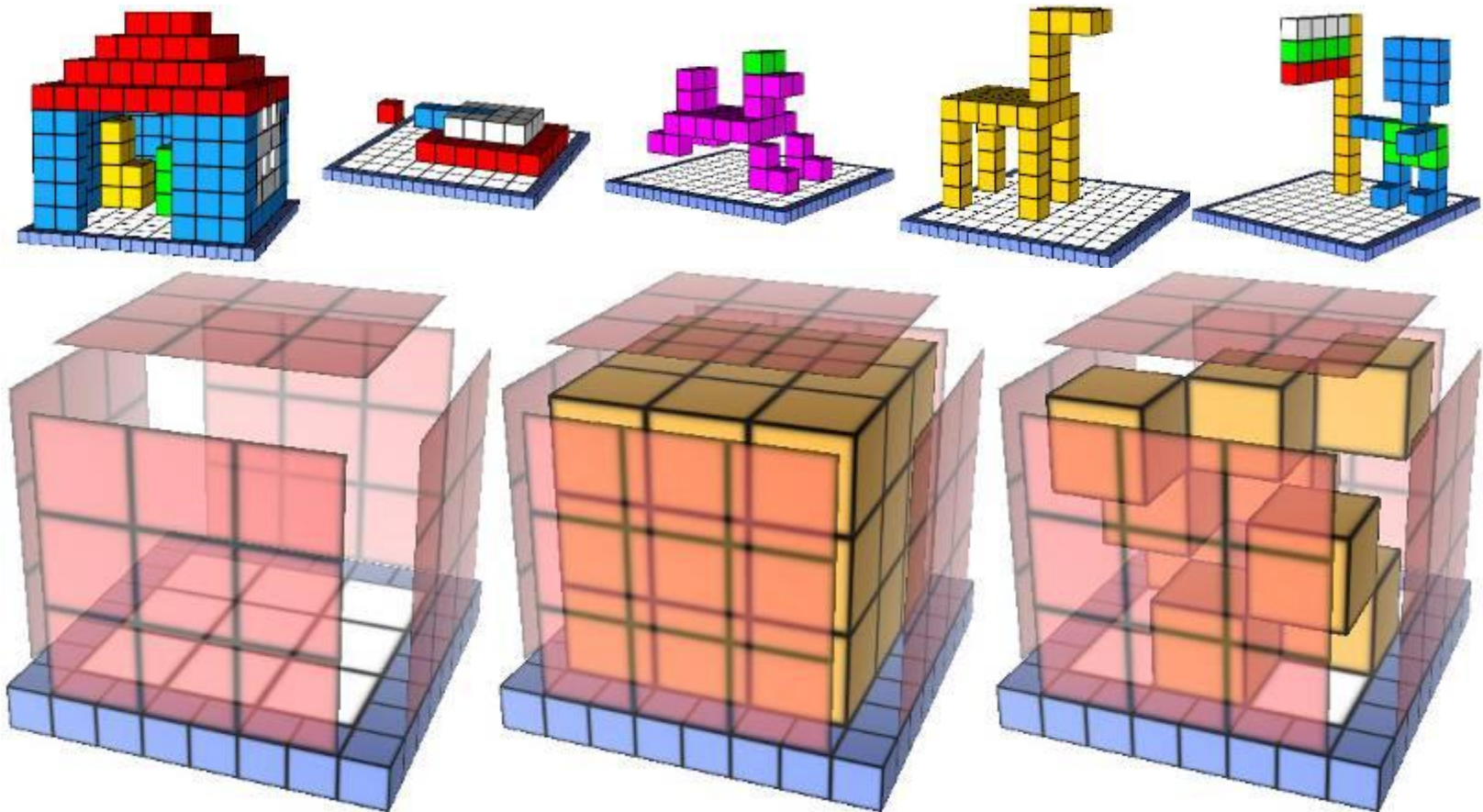


„Konstruktorius“



„Šešėliai“

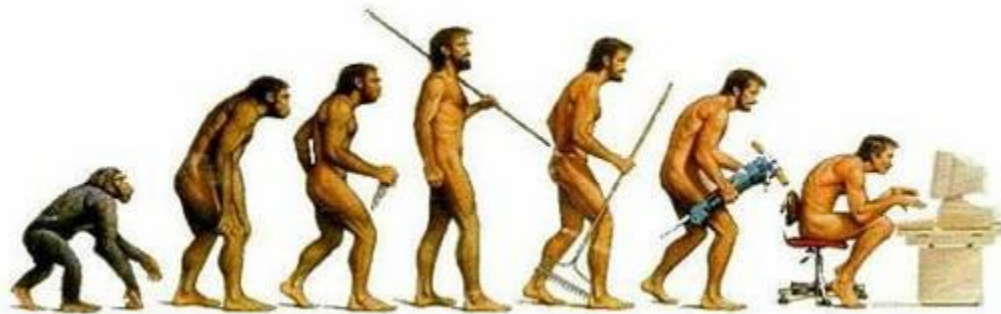
Fantazijos iš kubelių



TU ir IT

Pirmuosius dvejus metus mokome kalbėti ir vaikščioti, o kitus penkioliką – tylėti ir ramiai sėdėti!

Tai kaip jie gali išmokti naudotis IT?





- Atviro kodo programa [[nuoroda](#)]
- Skirta mokyti trimetės grafikos programavimo pagrindų
- Nereikia įdiegti į kompiuterį
- Kalba šiek tiek panaši į Java, C++ ir C#
- 3D judanti grafika
 - Daugiau kaip 20 scenų
 - Daug personažų
- Interaktyvios programos
- Yra pamokėlės pačioje Alice sistemoje
- Gana lengvai programuojama su paruoštais objektais (personažais)

Alice

The word "Alice" is written in a large, orange, 3D-style font. To its right is a small, cartoonish character of a blonde girl in a white dress and black shoes, holding a blue ball.

Objektinio programavimo aplinka



The screenshot displays the Alice software interface. At the top, the title bar reads "Alice (Storytelling Alice 10/01/2007) - C:\Documents and Settings\User\Desktop\Druskininkai\informacija\Alice\StorytellingAlice\Required\exampleWorlds\basketB...". The menu bar includes "File", "Edit", "Tools", and "Help".

The interface is divided into several panels:

- Objects:** A list of objects including World, Camera, Light, ground, basketball, Gym, gym, bleachers, and Amme. A "create new object" button is present.
- Scenes:** A 3D view of a gym scene with a character and a basketball. A "create new scene" button is at the top. The current scene is "Opening Scene Tripod".
- Events:** A panel for creating events. It shows a sequence: "When the world starts, do World.scene 1 method" and "Let move Camera".
- gym's details:** A panel with tabs for "properties", "methods", and "functions". It lists methods like "turn", "roll", "straighten up", "move", and "resize".
- Script Editor:** A panel for editing the "World.scene 1 method". It shows a sequence of actions: "Camera show title Bouncing a Basketball", "basketball move up 1.25 meters", "Amme touches the top side of the basketball", "Amme touch basketball side = up offset = 0", and a "Do together" block containing "Amme is going to keep touching the basketball while it moves around over the next 11 seconds" and "Amme keep touching basketball side = up offset = 0 duration = 11 seconds".

At the bottom of the script editor, there are buttons for "Do in order", "Do together", "If/Else", "Loop", "While", "For all in order", "For all together", "Wait", "print", and a comment symbol "//".

Programa primena struktūrograma



World.scene 1 method *No parameters* create new parameter

No variables create new variable

- Camera **show title** Cafeteria Romance *duration = 2 seconds* more...
- philip **say** Kristen, we are finally alone... more...
- philip.kiss
- LunchLady **say** What are you DOING? more...
- philip **turn to face** LunchLady more...
- Do together**
 - LunchLady.attempt to brainwash *who's head = philip.hips.lowerTorso.upperTorso.neck.head*
 - Do in order**
 - Wait 8 seconds
 - LunchLady **say** You do not like girls. *duration = 2 seconds* more...
- LunchLady **walk offscreen** *exit direction = left* more...
- philip **look** right more...
- Wait 0,5 seconds
- philip **look** left more...
- Wait 0,5 seconds
- philip **look** forward more...
- philip **say** What am I doing here? more...
- Do together**
 - Camera **get close up of** kristen *duration = 3 seconds* more...
 - philip **walk offscreen** *exit direction = right* more...
- kristen **look** down more...
- Camera **fade to black** more...

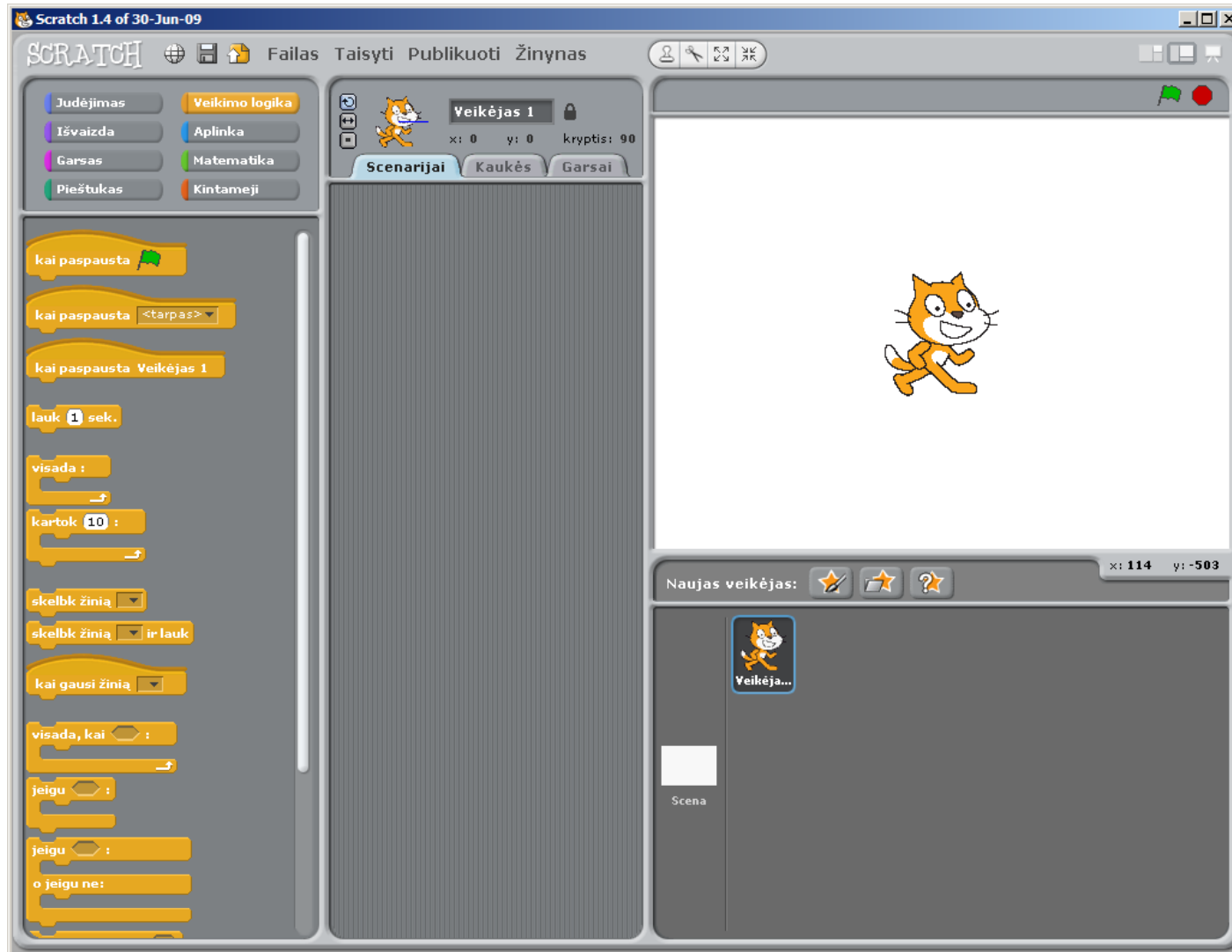
Do in order Do together If/Else Loop While For all in order For all together Wait print ⌘

Scratch



- Labai paprasta ir nepaprastai draugiška sistema
- Programa rašoma LEGO principu (struktūrogramų redaktorius)
- Lengvai suprantama
- Patogesnė negu Logo ir Komenskio Logo
- Kalba labai panaši į Logo
- Komandos paslėptos (jų nereikia rašyti ranka)
- Vaikas sprendžia konstravimo, bet ne gramatinių klaidų problemas
- Visiškai nemokama, laisvai platinama
- Jau tinka net 10-12 metų vaikams
- Lituanizuota, nors interneto svetainėje nėra lietuviškos medžiagos
- Darbus galima talpinti internete [[nuoroda](#)]

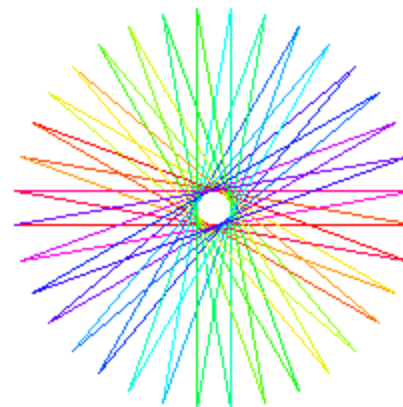
Scratch



Scratch



```
kai paspausta
išvalyk pašymus
nuleisk pieštuką
pasislėpk
pieštuko spalva = 0
kartok 36 :
  eik 200 žingsnius(-iu)
  pieštuko spalvą padidink 10
  pasisuk 170 laipsnių
```



Scratch komandų grupės



- Judėjimas
- Išvaizda
- Garsas
- Pieštukas
- Veikimo logika
- Aplinka
- Matematika
- Kintamieji

Judėjimas, Išvaizda, Garsas, Pieštukas

eik 10 žingsnius(-ių)

pasisuk ↻ 15 laipsnių

pasisuk ↻ 15 laipsnių

žiūrėk kryptimi 90°

žiūrėk į

eik į tašką x: 0 y: 0

eik ten, kur

slink 1 sek. į tašką x: 0 y: 0

prie x pridėk 10

x = 0

prie y pridėk 10

y = 0

jeigu priėjai kraštą, atšok

x koordinatė

y koordinatė

kryptis

pakeisk kaukę į kaukė2

kita kaukė

kaukės nr.

sakyk Labas! 2 sek.

sakyk Labas!

galvok Hmm... 2 sek.

galvok Hmm...

efektą spalva padidink 25

efekto spalva dydis = 0

panaikink grafinius efektus

padidėk 10 taškų

veikėjo dydis = 100 %

dydis

pasirodyk

pasislėpk

atsirask priekyje

pasitrauk atgal per 1 sloksnį(-ius)

grok miau

grok miau iki pabaigos

sustabdyti visus garsus

būgną 48 mušk 0.2 taktus

palauk 0.2 taktų/us

natą 60 grok 0.5 taktų/us

instrumentas = 1

pagarsink -10

garsumas = 100 %

garsumas

tempą padidink 20

tempas = 60 (taktai per minutę)

tempas

išvalyk paįšymus

nuleisk pieštuką

pakelk pieštuką

pieštuko spalva =

pieštuko spalvą padidink 10

pieštuko spalva = 0

pieštuko šešėlį padidink 10

pieštuko šešėlis = 50

pieštuko storį padidink 1

pieštuko dydis = 1

palik savo vaizdą (štampą)



Veikimo logika, Aplinka, Matematika, Kintamieji

The image displays a collection of Scratch code blocks organized into four columns:

- Action (Yellow):** Includes blocks for clicking (kai paspausta), pausing (lauk), repeating (kartok), scrolling (skelbk žinią), showing/hiding (kai gausi žinią), saying (visada, kai), if-then (jeigu), and ending (baigti šį scenarijų, sustabdyti viską).
- Environment (Blue):** Includes blocks for touching (lieti), asking (paklausk), answering (atsakymas), moving (pelės), clicking (paspausta), distance (atstumas iki), new (iš naujo paleisk laikmatį), time (laikmatis), coordinates (veikėjo), volume (garsumas), sound (yra garsas?), and sensitivity (slinktis, jutklis).
- Math (Green):** Includes arithmetic (+, -, *, /), range (bet kuris sk. nuo), comparison (<, =, >), logical (ir, arba, ne), concatenation (sujunk), string (rytas), length (rytas ilgis), division (iš dalybos liekana), and square root (kvadratinė šaknis iš).
- Variables (Orange):** Includes creating (Sukurti kintamąjį, Šalinti kintamąjį), checkboxes (a, b), assignment (a = 0), addition (a pridėk), display (parodyk), increment (paslėpk), creating (Sukurti sąrašą, Šalinti sąrašą), checkbox (s1), list operations (i sąrašą pridėk, iš s1 ištrink nr., i s1 vieta įterpk, pakeisk nr. i), and list display (sąrašo nr., sąrašo ilgis, sąrašo yra daiktas?).



Squeak

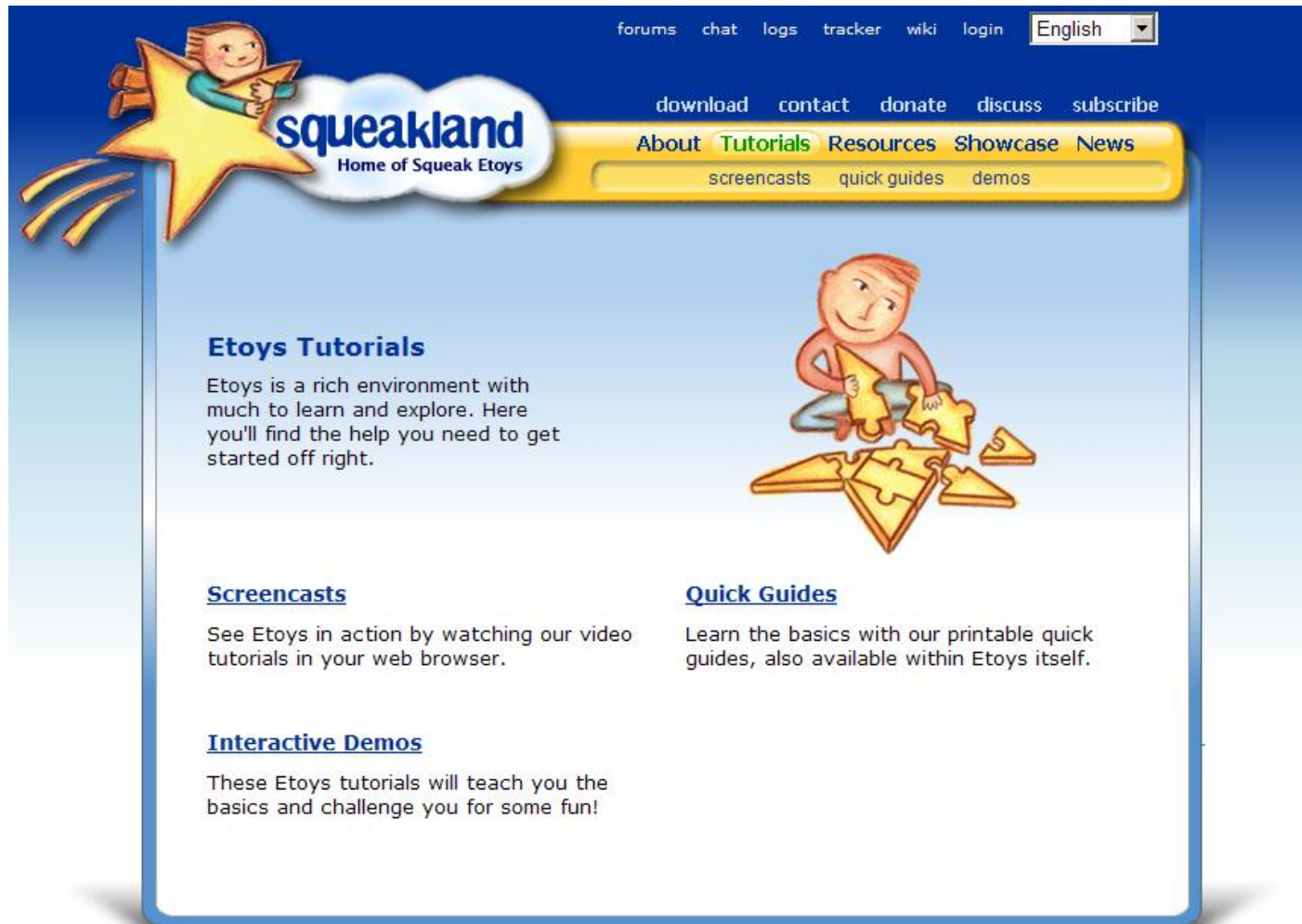


- Remiasi Smalltalk kalba, kuri sukurta prieš 35 metus panaudojant Simula, Sketchpad ir Lisp idėjas
- Galima kurti interaktyvias programas
- Programavimo kalba paprasta, bet programuoti sudėtinga
- Nemokama
- Nereikia diegti (atsisiuntei ir dirbi)
- Miniatiūrinė programavimo sistema
- Muzika ir garsai realiaame laike
- Veikia Windows, CE/PocketPC, MacOS, Linux/Unix, OS/2 Warp, Acorn RiscOS sistemose
- Sukūrė Rob Pike & Luca Cardelli 1985m.

Squeak nuorodos



- SqueakLand - mokiniamas
- Squeak Smalltalk – informacija
- <http://www.squeakland.org/tutorials/demos/> - Interaktyvūs pavyzdžiai



forums chat logs tracker wiki login English

download contact donate discuss subscribe

About **Tutorials** Resources Showcase News

screencasts quick guides demos

Etoys Tutorials

Etoys is a rich environment with much to learn and explore. Here you'll find the help you need to get started off right.

Screencasts

See Etoys in action by watching our video tutorials in your web browser.

Interactive Demos

These Etoys tutorials will teach you the basics and challenge you for some fun!

Quick Guides

Learn the basics with our printable quick guides, also available within Etoys itself.

Squeak



Etoys Navigation

Navigator Bar: Keep a Project and Find It Again

Click the journal icon in your Navigator Bar to keep a project.



Etoys Painting

Paint Tools: Six Sizes of Brushes, Erasers, and Clean Edge Tools



Etoys Halo

Halo Tools: How to Make the Halo Handles Show



Here are halo handles around a green rectangle. The halo is a set of controls that let you make changes to objects in Etoys.

Right click on the rectangle to open a usable halo called "rectangled".

Etoys Supplies

Supplies: The Object Catalog Contains Tools and Objects Ready to Use



Etoys Object Catalog

Object Catalog: Slider for Remote Control



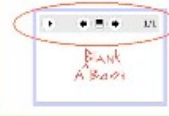
A slider bar can be scripted to control other objects.

Etoys Books

Books: Five Top Border Icons



Click on Supplies and then on the book there.



Etoys Tiles

Script Tiles: Forward by 5, a Tile That Makes an Object Move.

Open Supplies and click on an Ellipse.

Open a Viewer for the Ellipse with its halo's cyan eye.



Etoys Menus

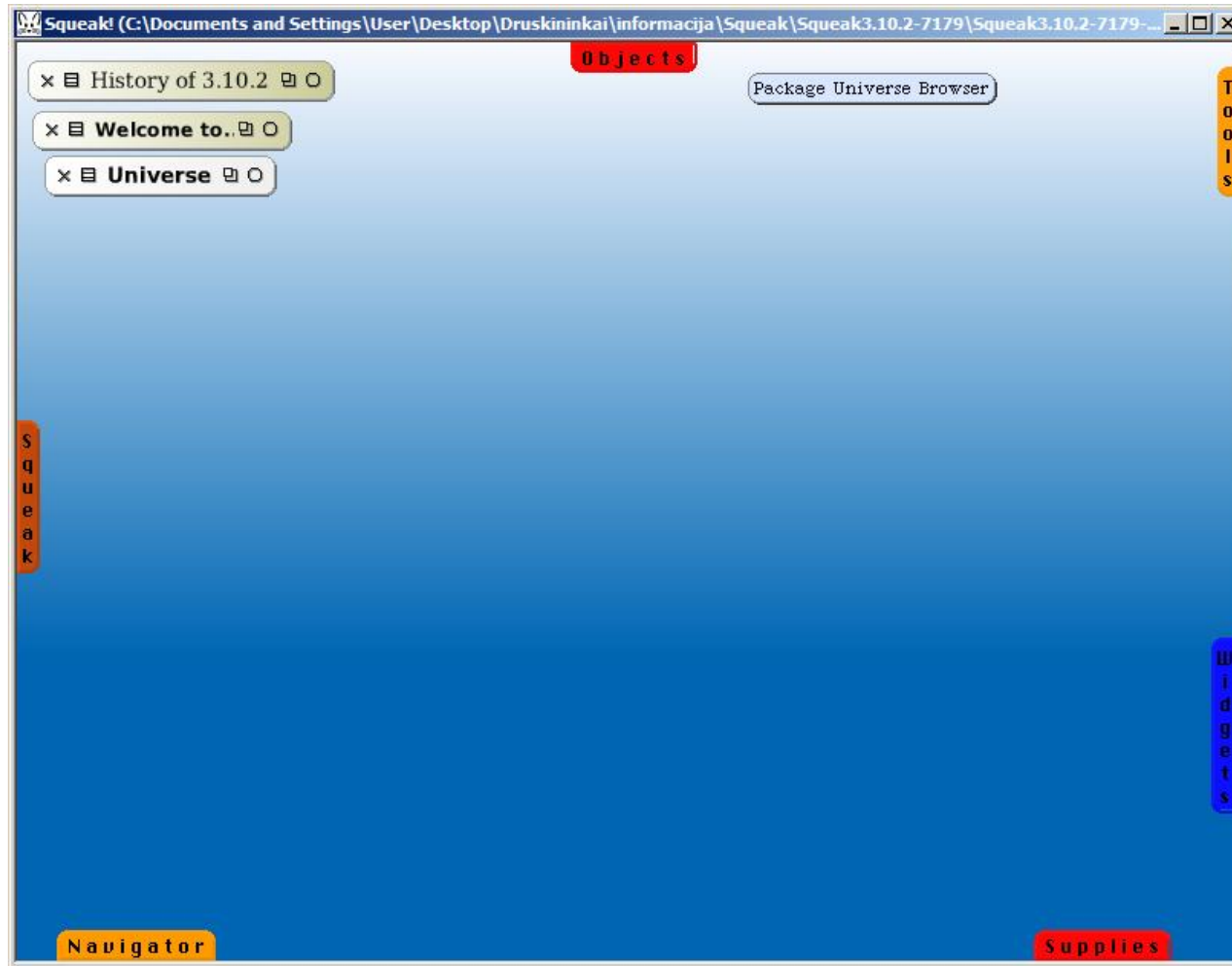
Menu Tools: What Is Named or When Should This Script Run?

Click the green check to start the script ticking and click again to pause.



```
brush script | [control]
brush forward by [random] (100)
brush turn by [random] (80)
```

Squeak neįprasta darbo aplinka



Paskalio šeima

- Pascal (1970 m.)
- Turbo Pascal (1986 m.)
- Oberon (1985 m.)
- Delphi terpė (1995 m.)
- Free Pascal (1993-2000 m.)
- Component Pascal (1997m.)
- Lazarus terpė (2006-2008 m.)
- BlackBox terpė
- CPlde terpė
- OpenDelphi.org! Ar turėsime?

Free Pascal

- fpc-2.2.2.x86_64-win64.exe (2008-08-03)
- Tradicinė „Turbo“ aplinka
- Ar dar yra klaidų kompiliatoriuje?

Component Pascal

- Kūrėjas „Oberon Microsystems AG“
- Programų saugumas
- Nereikia didelių išteklių
- Mažai užimanti sistema
- Atsisakyta GOTO
- Griežta kontrolė
- Viskas, kas įmanoma iškelta į bibliotekas
- Kitoks požiūris į įvedimą-išvedimą

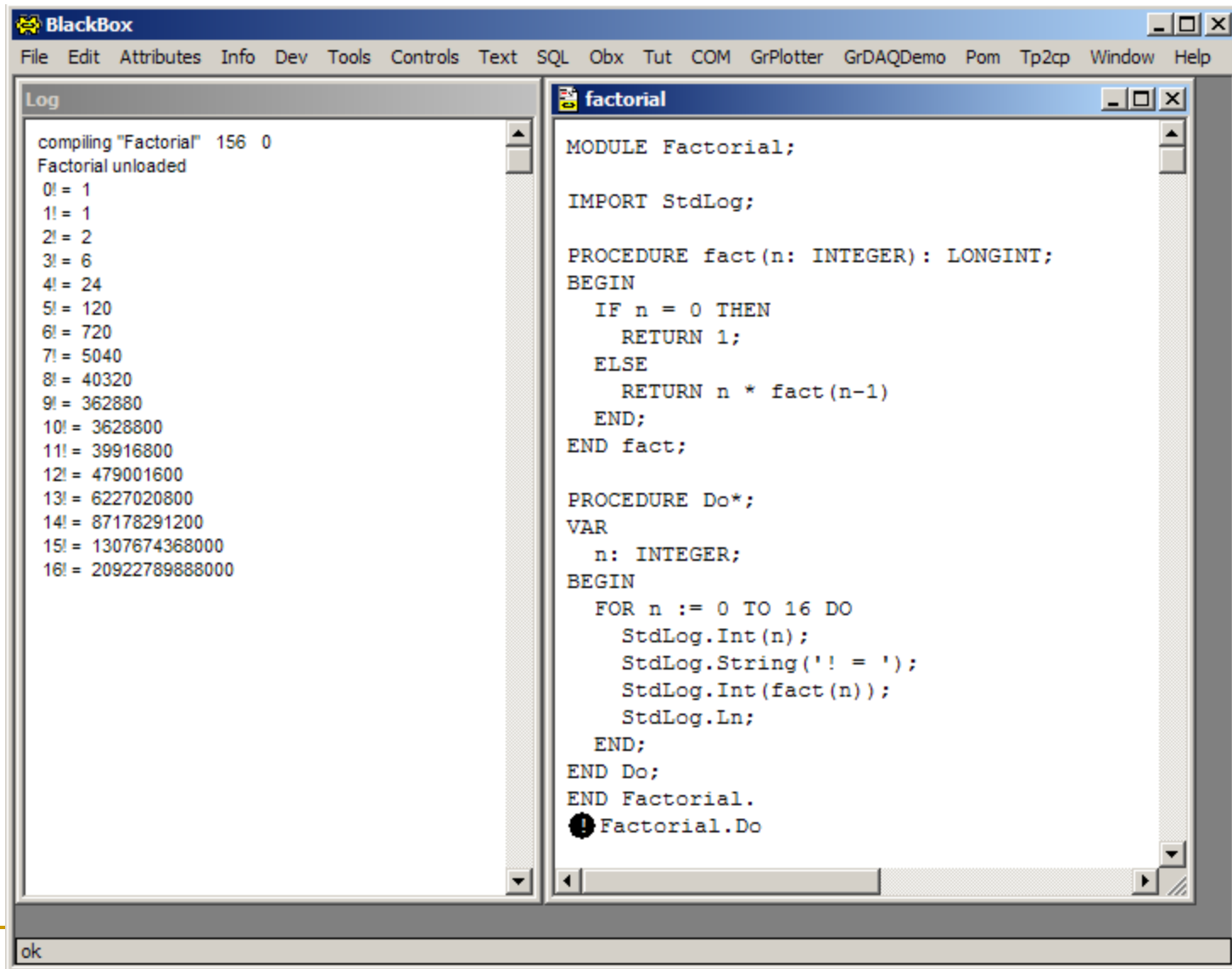
Component Pascal

- Minimali, paprasta ir greitai išmokstama sintaksė
- Atsisakyta retai naudojamų tipų
- Supaprastinta masyvo samprata
- Nėra vykdymo žingsniais, tačiau yra kitos programos veikimo teisingumo tikrinimo priemonės (pvz. ASSERT)
- Plačiau <http://aldona.mii.lt/pms/lok/bb/BB.htm>
- Sintaksė ir skirtumai

Paprasciausia programa

```
MODULE AsEsu;  
IMPORT StdLog;  
  PROCEDURE Daryk*;  
    VAR  
      a, b, c : INTEGER;  
  BEGIN  
    StdLog.String('Aš esu Oberono sūnus!');  
    StdLog.Ln;  
  END Daryk;  
END AsEsu.  
● AsEsu.Daryk
```

BlackBox aplinka



The screenshot shows the BlackBox software interface. The main window has a menu bar with options: File, Edit, Attributes, Info, Dev, Tools, Controls, Text, SQL, Obx, Tut, COM, GrPlotter, GrDAQDemo, Pom, Tp2cp, Window, Help. There are two main panes:

- Log pane:** Displays the output of a factorial program. It starts with "compiling 'Factorial' 156 0" and "Factorial unloaded". Below that, it lists factorial values from 0! to 16!.
- factorial pane:** Contains the source code for a factorial program. The code defines a module, imports StdLog, and includes a recursive fact procedure and a Do* procedure that loops from 0 to 16, printing each factorial value.

```
compiling "Factorial" 156 0
Factorial unloaded
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
11! = 39916800
12! = 479001600
13! = 6227020800
14! = 87178291200
15! = 1307674368000
16! = 20922789888000

MODULE Factorial;

IMPORT StdLog;

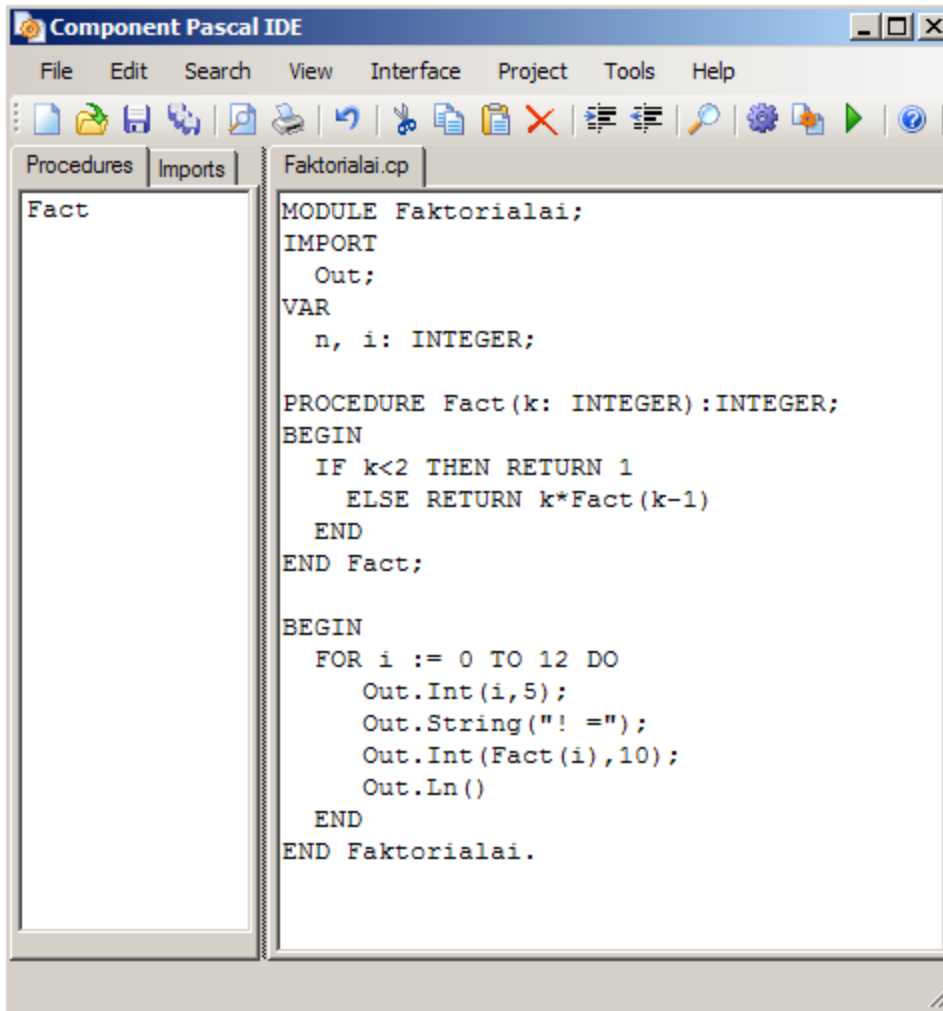
PROCEDURE fact(n: INTEGER): LONGINT;
BEGIN
  IF n = 0 THEN
    RETURN 1;
  ELSE
    RETURN n * fact(n-1)
  END;
END fact;

PROCEDURE Do*;
VAR
  n: INTEGER;
BEGIN
  FOR n := 0 TO 16 DO
    StdLog.Int(n);
    StdLog.String('! = ');
    StdLog.Int(fact(n));
    StdLog.Ln;
  END;
END Do;
END Factorial.
! Factorial.Do
```

ok

CPIde terpė

<http://www.cfbsoftware.com/cpide/cpide.aspx>



The screenshot shows the Component Pascal IDE window titled "Component Pascal IDE". The menu bar includes File, Edit, Search, View, Interface, Project, Tools, and Help. Below the menu is a toolbar with various icons. The main editing area is divided into two panes: "Procedures" on the left and "Imports" on the right. The "Procedures" pane contains a tree view with a single entry "Fact". The "Imports" pane is empty. The main code editor displays the following Pascal code:

```
MODULE Faktorialai;
IMPORT
  Out;
VAR
  n, i: INTEGER;

PROCEDURE Fact(k: INTEGER):INTEGER;
BEGIN
  IF k<2 THEN RETURN 1
  ELSE RETURN k*Fact(k-1)
  END
END Fact;

BEGIN
  FOR i := 0 TO 12 DO
    Out.Int(i,5);
    Out.String("! =");
    Out.Int(Fact(i),10);
    Out.Ln();
  END
END Faktorialai.
```

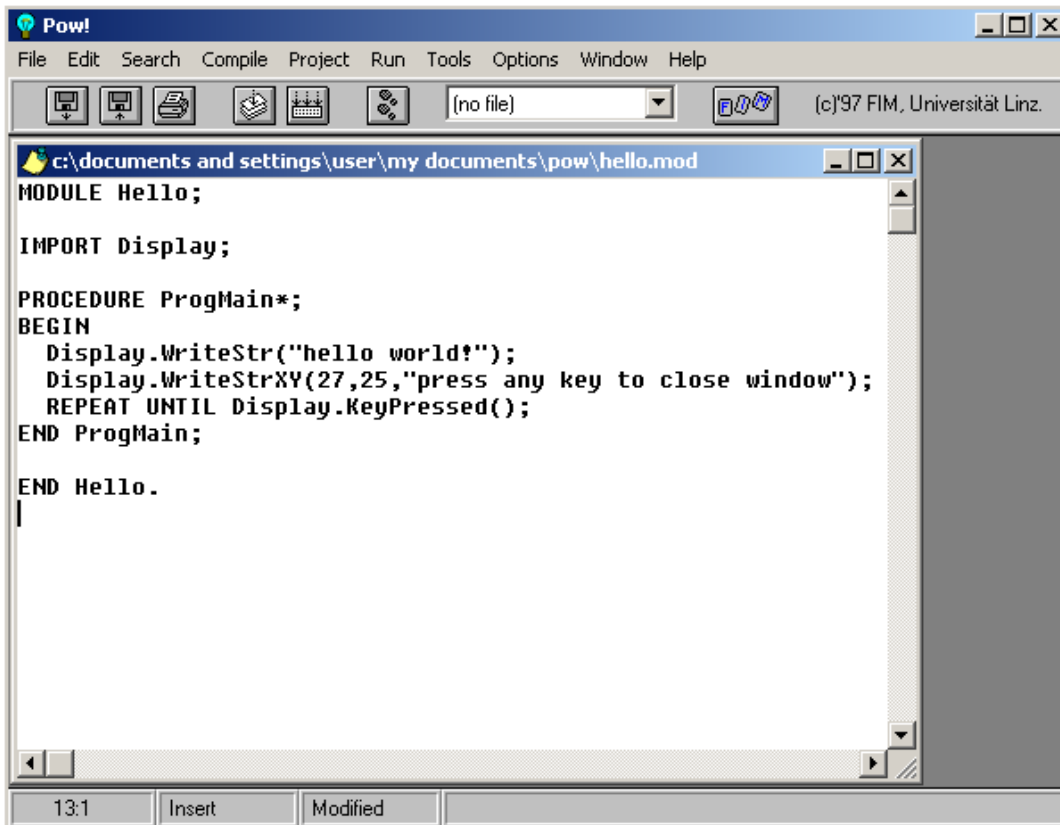
- Patogi tradicinė aplinka
- Paprasta naudoti
- Laisvai platinama
- Kompiluoja .exe ir .dll bylas

Gardens Point Component Pascal GPCP kompilatorius

- Kompilatorius CPlde terpei
- Galima kompiliuoti tiesiai interneto svetainėje
 - http://plas.fit.qut.edu.au/gpcp/gpcp_online.aspx
- Terpei ir vykdymui reikalingi
 - Microsoft .NET Framework version 2.0 Redistributable Package
 - Microsoft .NET Framework Version 2.0 Software Development Kit (SDK)
 - GPCP kompilatorius

<http://plas.fit.qut.edu.au/gpcp/Downloads.aspx>

POW! – Programmers Open Workbench

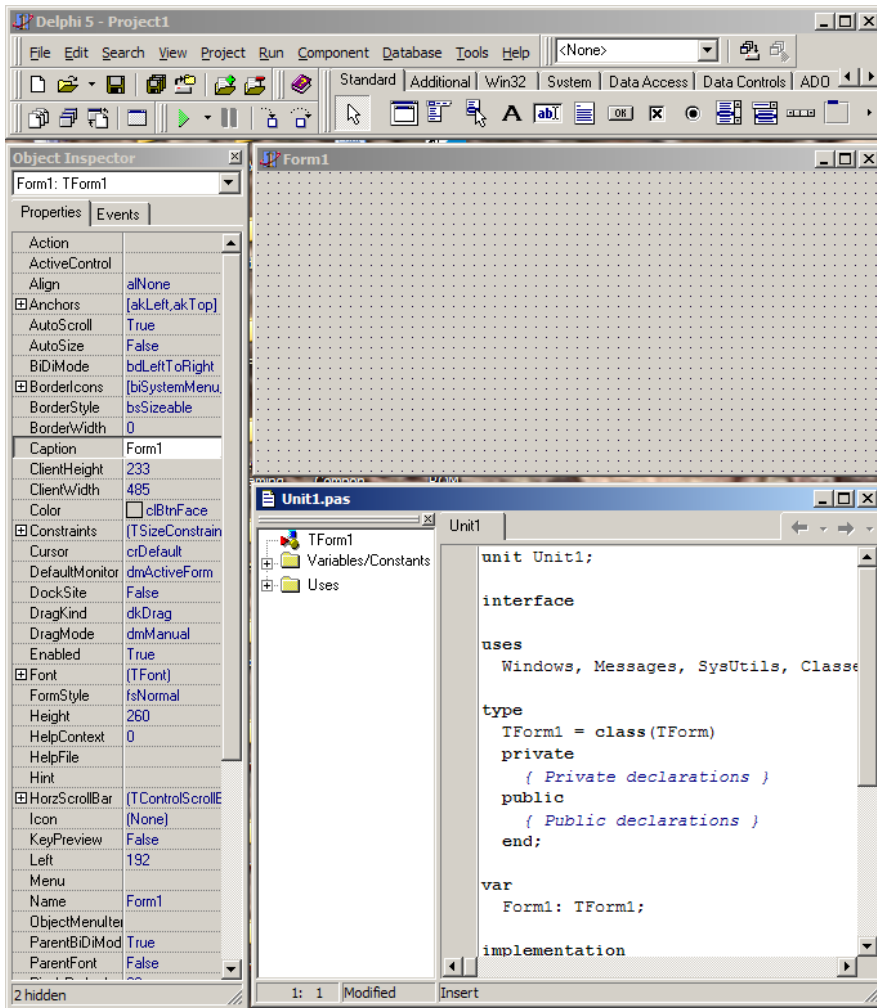


```
MODULE Hello;
IMPORT Display;
PROCEDURE ProgMain*;
BEGIN
  Display.WriteStr("hello world!");
  Display.WriteStrXY(27,25,"press any key to close window");
  REPEAT UNTIL Display.KeyPressed();
END ProgMain;
END Hello.
```

- Oberon-2
- Java
- C/C++

<http://www.fim.uni-linz.ac.at/Research/POW/>

Delphi



- Komercinė programa
- Objektinio programavimo kalba
- Vizualios kūrimo priemonės
- Greitas kompiliavimas
- Jau pritaikyta .NET platformai
- Galima greitai ir patikimai programuoti Windows aplinkoje

Lazarus

- Laisvai platinama
- Sąsaja labai panaši į Delphi
- Objektinio programavimo kalba
- Vizualios kūrimo priemonės
- Galima greitai ir patikimai programuoti Windows aplinkoje
- Yra klaidų, būdingų Free Pascal
- Kompiliuojamas kodas yra nepaprastai didelės apimties, atjungus kontrolę – sumažėja
- Nėra pilnai suderinamas su Delphi (Delphi => Lazarus)

OpenDelphi.org

- Turėtų būti laisvai platinama
- Turėtų būti pilnai kontroliuojama Borlando kompanijos

<http://opendelphi.org/>

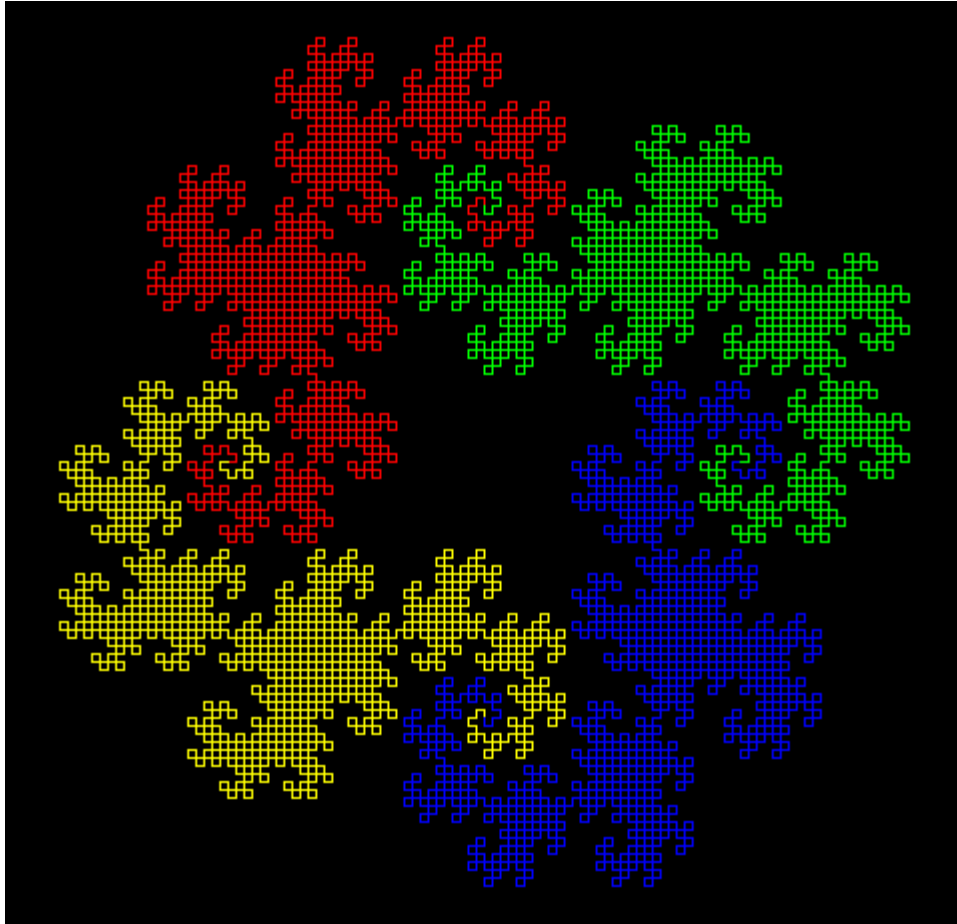


C kalbos šeima

- Dev C++
- C++ Builder (1997 m.)
- Visual C++ Express Edition

DEV-C++ (Bloodshed Software)

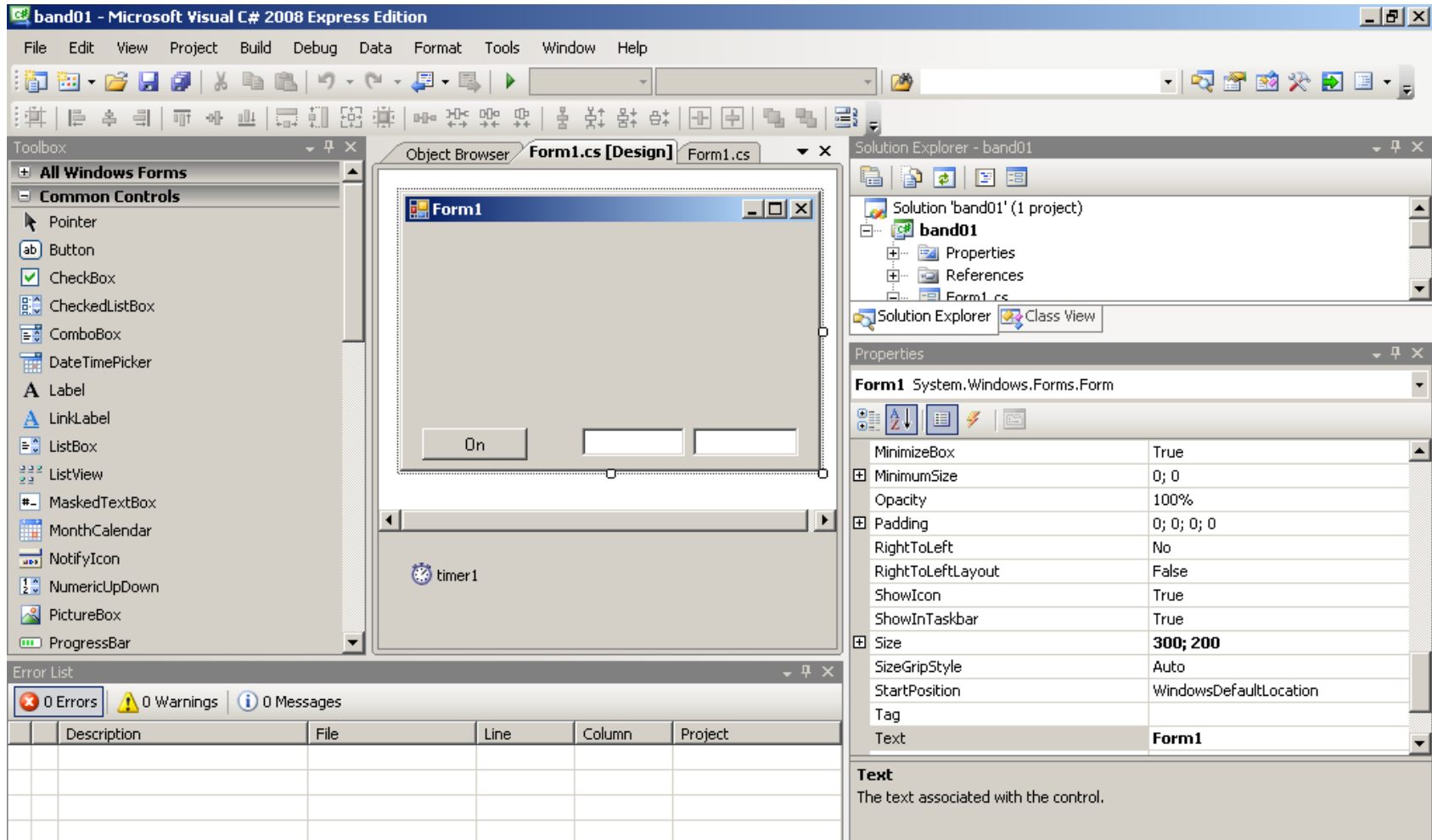
- Galima kurti:
 - Windows programas
 - Konsolines programas
 - Statines bibliotekas
 - DLL bibliotekas
- Nedidelė ir patogi mokytis programuoti
- Michael Main sukūrė grafikos biblioteką
 - graphics.h; libbgi.a



C++ Builder (Borland Software corporation)

- Komerčinė
- Greito projektavimo sistema
- Vizualus projektavimas
- Terpė analogiška Delphi
- Galima naudoti daugelį Delphi komponentų

Visual C++ Express Edition (Microsoft)



Visual C++ Express Edition

- Vizualus programavimas
- Nemokama
- Populiari studentų tarpe
- Apkarpyta „Microsoft Visual Studio“
- Objektinio programavimo lygis
- 2D ir 3D grafika
- Tinklapis:
 - <http://www.microsoft.com/express/vc/>
 - <http://www.microsoft.com/express/download/>

Visual Studio Express Edition sudėtinės dalys

- Visual Basic Express Edition
- Visual C# Express Edition
- Visual C++ Express Edition
- Visual J# Express Edition
- Visual Web developer Express Edition
- SQL Server Express Edition

Išvados

- Programavimo kalbų pasirinkimas didelis
- Programavimo tikslų pasirinkimas:
 - Programavimo pradmenų mokymas
 - Ruošimasis programavimo olimpiadoms
 - Taikomųjų programų kūrimas vizualaus programavimo sistemomis
 - Procesų modeliavimas (panaudojant ir grafines priemones)

Programavimo pradmenų mokymas

- Komenskio Logo
- Imagine Logo
- Elica
- Scratch
- Free Pascal
- DEV-C++
- BlackBox
- CPlde
- Oberon

Kaip parinkti programavimo kalbą?

- Ko mes tikimės iš programavimo kalbos?
- Kiek pastangų reikės, kad mokinys suprastų, jog jau programuoja?
- Ar po to jam kils noras pasidžiaugti su kitais savo pasiekimais?
- Ar programavimo kalba gerai struktūrizuota ir neturi nereikalingo balasto?
- Kiek išmoktos konstrukcijos bus funkcionalios?
- Kiek pastangų turės įdėti mokytojas, mokydamasis naujos programavimo kalbos?
- Ar mes norime „industrinės“ programavimo kalbos, ar išmokinti programavimo pagrindų?
- Kiek yra literatūros apie pasirinktą programavimo kalbą?
- Išmokti programuoti yra sunku

Ruošimasis programavimo olimpiadoms

- DEV-C++
- Free Pascal

Taikomųjų programų kūrimas vizualaus programavimo sistemomis

- Lazarus
- Visual C++ Express Edition
- Delphi
- C++ Builder
- Ultimate++

Procesų modeliavimas

- Elica
- Imagine Logo
- C++ Builder + OpenGL
- Lazarus + OpenGL
- Delphi + OpenGL
- Visual C++ Express Edition

Programavimo pradmenų mokymui, kai norima pereiti prie industrinės kalbos

- Free Pascal/ Oberon/ Komp. Paskalis
 - Paprasta, griežta ir nesunkiai išmokstama kalbos sintaksė
 - Pratinimas naudotis bibliotekomis
 - Pratinimas dokumentuoti programas
 - Pratinimas išmąstyti algoritmą
 - Pratinimas kurti programą priartėjimų metodu

Programavimo pradmenų mokymui, kai norima pereiti prie industrinės kalbos

■ C/C++ šeimos kalbos

- ❑ Kruopščiai parinkti kalbos poaibį su paprasta, griežta ir nesunkiai išmokstama kalbos sintakse (kol kas to dar niekam nepavyko padaryti)
- ❑ Parinkti kalbos poaibio mokymo strategiją ir taktiką
- ❑ Atrinkti būtiniausias bibliotekas, pratinti jomis naudotis
- ❑ Pratinti dokumentuoti programas
- ❑ Pratinti išmąstyti algoritmą
- ❑ Pratinti kurti programą priartėjimų metodu
- ❑ Intensyviai mokytis skaityti programas

D programavimo kalba

- <http://www.digitalmars.com/d/2.0/index.html>
- C++, Python, Ruby kalbų geriausios savybės
- Ar neatsitiks kaip su OS Oberon?

Daugiau informacijos

- <http://popmokslas.projektas.lt/failai/lpm/html/lt/index.html>
- <http://www.akl.lt/programos>
- [http://www.ultimatepp.org/www\\$suppweb\\$overview\\$en-us.html](http://www.ultimatepp.org/www$suppweb$overview$en-us.html)

Pabaiga

Kad ir kokią egzotišką programavimo kalbą (pvz. EIFFEL) pasirinktumėm, turime atsižvelgti į mokytojo kompetenciją. Tik remiantis jo kompetencija galima rasti sėkmingą kelią

Ar ši kalba tinka pradžiamoksliui?

indexing

```
description: "Programa Labas Paryžiau!"  
author: "Albertas Dinda"
```

class

```
HELLO
```

creation

```
make
```

feature

```
make is
```

```
-- paprastas spausdinimas
```

```
do
```

```
    io.put_string ("Salut, Paris!")
```

```
    io.put_new_line
```

```
end
```

```
end -- class HELLO
```

Ar ši kalba tinka pradžiamoksliui?

```
class HELLO
  creation make
  feature make is
    do print ("Salut, Paris!") end
end
```

Ar ši kalba tinka pradžiamoksliui?

```
#include <iostream>
// šito reikia dėl std::cout ir std::endl

int main()
{
    std::cout << "Salut, Paris!";
    std::cout << std::endl;
    return 0;
}
```

Ar ši kalba tinka pradžiamoksliui?

```
MODULE LabasParyziau;  
  IMPORT Log;  
  PROCEDURE Daryk*;  
  BEGIN  
    Log.String("Salut, Paris!");  
    Log.Ln  
  END Daryk;  
END LabasParyziau.
```

Ar ši kalba tinka pradžiamoksliui?

```
TO HELLOPARIS  
  PRINT [Salut, Paris!]  
END
```

Ar ši kalba tinka pradžiamoksliui?

```
Program HelloParis (output) ;  
begin  
    writeln ('Salut, Paris! ');  
    readLn ;  
end.
```

Ar ši kalba tinka pradžiamokslui?

```
public class HelloParis {  
    public static void main(String[] args) {  
        System.out.println("Salut, Paris!");  
    }  
}
```

Ar ši kalba tinka pradžiamoksliui?

```
#!/usr/bin/perl  
print "Salut, Paris!\n";
```

Ar ši kalba tinka pradžiamokslui?

```
assume cs:cseg, ds:dseg, ss:sseg  
mov ax,dseg  
mov ds,ax ;duomenu segmento iniciavimas  
lea dx, welcome  
mov ah, 09h  
int 21h  
mov ah,4ch  
int 21h
```

Gal šios kalbos tinka pradžiamokslui?

```
10 REM PRANESIMAS
20 PRINT "Salut, Paris!\n"
30 RET
```

```
10.05 C pranesimas
10.10 T "Salut, Paris!" !
10.15 R
```

```
program Salut
print *, "Salut, Paris!"
end
```

Gal šios kalbos tinka pradžiamokslui?

Проц Старт()

Вывод 'Здравствуй, Новосибирск!!!'

Кон Проц

```
>>> print («Salut, %s!» % "Python")
```

```
." Salut, FORTH"
```

Geriausiai tiks ta
programavimo kalba,
kurią geriausiai moka
mokytojas