

GRAFIKA W JĘZYKU PROGRAMOWANIA

Data przeprowadzenia lekcji: 09.04.2013r.

Nauczyciel prowadzący: mgr inż. Wiesława Kozłowska

Konspekt lekcji do przedmiotu: Programowanie strukturalne i obiektowe, mający na celu pokazanie uczniom różnicy między tworzeniem obrazu w programie graficznym a programowaniem grafiki przy wykorzystaniu języka programowania.

SCENARIUSZ LEKCJI OPARTEJ NA PROGRAMIE DLA KLASY II TECHNIKUM W ZAWODZIE TECHNIK INFORMATYK

Nr.prog.312[01]/T, SP/MENIS/2004.06.14

Przedmiot: Programowanie strukturalne i obiektowe

Dział programowy: Podprogramy

Temat: Grafika w języku programowania

Klasa: II G

Czas trwania lekcji: 45 min

CELE OGÓLNE:

- pokazanie uczniom różnicy między tworzeniem obrazu w programie graficznym a programowaniem grafiki przy wykorzystaniu języka programowania
- wykorzystanie przez ucznia poznanych już zasad programowania, konstrukcji
- programów, zwłaszcza technik algorytmicznych i definiowanie własnych procedur

CELE OPERACYJNE:

WIADOMOŚCI:

A. Zapamiętanie wiadomości:

- podać zadeklarować prawidłowo moduł „uses graph”
- podać procedury: InitGraph, CloseGraph, ClearDevice, repeat until KeyPressed
- podać procedury:
 - PutPixel (x, y, k),
 - Line(x1, y1, x2, y2),
 - MoveTo (x, y),
 - LineTo(x, y),
 - Rectangle (x1, y1, x2, y2),
 - Circle(x, y, r),
 - GetPixel (x, y),
 - SetColor (k),
 - SetBkColor,
 - SetTextStyle,
 - SetFillStyle (p, c),
 - Bar(x1, y1, x2, y2),
 - FloodFill (x, y, k)
 - GetMaxX, GetMaxY

B. Rozumienie wiadomości:

- uruchomić program
- zastosować procedury graficzne we Free Pascalu
- przejść do trybu graficznego we Free Pascalu

UMIEJĘTNOŚCI**C. Stosowanie wiadomości w sytuacjach typowych:**

- potrafi prawidłowo stworzyć program stosując procedury graficzne
- potrafi zdefiniować proste procedury graficzne we Free Pascalu
- potrafi korzystać z instrukcji pomocniczej rysować punkty, proste, krzywe, figury oraz wypełnić barwą wyznaczony obszar we Free Pascalu
- potrafi zaplanować działania prowadzące do napisania prostego programu z wykorzystaniem procedur graficznych

D. Stosowanie wiadomości w sytuacjach problemowych:

- dokonuje analizy przykładowych programów zawierający procedury graficzne
- prezentuje własne rozwiązania problemów w twórczy sposób i dzielić się wiedzą z innymi

METODY I FORMY PRACY:

- pokaz
- dyskusja
- ćwiczenia
- praca indywidualna

ŚRODKI DYDAKTYCZNE

- pracownia komputerowa
- oprogramowanie: Windows 7, Office 2007, Free Pascal
- instrukcje do ćwiczeń
- prezentacja elektroniczna

PRZEBIEG LEKCJI

- Sprawdzenie obecności. Podanie tematu i celu zajęć. Pozostawienie uczniom czasu na przygotowanie się do lekcji (5 min)

WPROWADZENIE

- Przypomnienie wiadomości z ostatniej lekcji (5 min)

FAZA REALIZACJI (30 min)

Nauczyciel omawia, w jaki sposób wypełnić barwą wyznaczony obszar.

- Uczniowie, korzystając z instrukcji do ćwiczeń wykonują program rysujący na środku ekranu żółte koło o promieniu 100 wypełnione pogrubionymi liniami pochyłymi w prawo.

- Uczniowie, korzystając instrukcji do ćwiczeń wykonują program rysujący na ekranie monitora flagę Szwajcarii (biały krzyż na czerwonym tle).
- Uczniowie samodzielnie piszą program rysujący zielony prostokąt wypełniony kropkami w lewym górnym rogu ekranu.

UTRWALENIE WIEDZY (5 min)

- A. Nauczyciel zadaje pytania kontrolne
- B. Zakończenie zajęć:
 - wylogowanie z systemu,
 - uporządkowanie stanowisk pracy,
 - pożegnanie uczniów

ĆWICZENIA NA LEKCJĘ:

Zad 1.

Napisz program rysujący na środku ekranu żółte koło o promieniu 100 wypełnione pogrubionymi liniami pochyłymi w prawo.

Zad. 2.

Napisz program rysujący na całym ekranie monitora flagę Szwajcarii (biały krzyż na czerwonym tle).

Zad. 3.

Napisz program rysujący zielony prostokąt wypełniony kropkami otaczający w lewym górnym rogu ekranu.

Temat: GRAFIKA W JĘZYKU PROGRAMOWANIA

1. Podstawowe pojęcia związane z grafiką

Do włączenia trybu graficznego służy procedura **INITGRAPH**. Aby program mógł używać grafiki musi się na początku znajdować się zapis **USES GRAPH**. Katalog do grafiki na dysku C musi zawierać sterowniki kart graficznych (C:\TP\BGI). Po zakończeniu pracy z grafiką, należy wrócić z grafiki do tekstu procedurą **CLOSEGRAPH**. Aby zapoznać się z efektem swojej pracy, należy na końcu zastosować instrukcję **REPEAT UNTIL KEYPRESSED** – oczekiwanie na naciśnięcie dowolnego klawisza.

Wywołanie procedury INITGRAPH ma następujące parametry:

INITGRAPH(karta graficzna, tryb, ścieżka dostępu do sterownika);

gdzie zmienne typu karta i tryb są przekazywane przez typ integer,

np. INITGRAPH(ster, tryb, 'C:\TP\BGI\');

przykład:

```
program grafika;
uses crt, graph;
var ster, tryb:integer;
BEGIN
ster:=detect; {wykrywanie trybu pracy karty graficznej}
initgraph (ster, tryb, 'C:\TP\BGI\');
... {instrukcje graficzne i nie tylko}
repeat until keypressed;
closegraph; {powrót do trybu tekstowego}
END.
```

2. Budowa ekranu

Ekran w trybie graficznym składa się z pikseli (czyli jest to punkt, najmniejszy, niepodzielny element obrazu). Każdy punkt ma swoje współrzędne:



3. Omówienie procedur graficznych

- **PUTPIXEL (x, y, k)** – rysuje punkt w podanym miejscu na ekranie w zadanym kolorze, x, y – współrzędne, k – kolor (0-15)
przykład: putpixel (100, 100, 15) – punkt o współrzędnych 100,100 w kolorze białym
- **LINE (x1, x2, y1, y2)** – rysuje odcinek między punktami (x1, y1) (x2, y2)
przykład: line (1,1,639,1) – rysuje poziomą linię na górze ekranu
- **MOVETO (x, y)** – ustala pozycję początku odcinka dla procedury w punkcie (x, y) i przenosi punkt w dane miejsce na ekranie
przykład: moveto (100,100)
 lineto (150, 150)
 lineto (200,100)
 lineto (250,50)
 lineto (300,100)
 lineto (350,50)
 lineto (400,100) – rysuje zygzak na górze ekranu
- **LINETO (x, y)** – przesuwa kursor. Ma dwa parametry. Zaczyna rysować od kursora graficznego. Po narysowaniu przenosi kursor na jego koniec.
- **LINEREL (x, y)** – dzielenie niewielkiej figury na ekranie na kilka miejsc
- **RECTANGLE (x1, y1, x2, y2)** – rysuje prostokąt o bokach równoległych do osi OX i OY, którego lewy górny i prawy dolny wierzchołek mają odpowiednie współrzędne
przykład: rectangle (1,1,319,239)
- **CIRCLE (x, y,r)** – rysuje okrąg o środku w punkcie (x, y) i promieniu r (wyrażonym w pikselach)
przykład: circle (319, 239, 100) – rysuje okrąg na środku ekranu o promieniu 100 pikseli
- **ARC (x, y, a1, a2, r)** – rysuje łuk jako wycinek okręgu o środku w punkcie (x, y) i promieniu r oraz kącie początkowym (a1) i końcowym (a2) wyrażonym w stopniach
przykład: arc (319, 239, 0, 90, 150) – rysuje łuk w prawej górnej ćwiartce ekranu
- **ELIPSE (x, y, a1, a2, rx, ry)** – rysuje wycinek elipsy o środku w punkcie (x, y) i promieniu rx i promieniu pionowym ry zaczynając od punktu odchylenia) o kat a1 i kończąc na punkcie odchylonym o kat a2. przykład: ellipse (159, 239, 0, 360, 159, 239) - rysuje elipsę w lewej połowie ekranu
- **CLEARDEVICE** - procedura czyści ekran graficzny.

RYSOWANIE PUNKTÓW NA EKRANIE:

Każdy punkt określa para współrzędnych typu WORD. Odpowiedzialne za to są dwie funkcje:

- GetMaxX (639)
- GetMaxY (349)

RYSOWANIE LINII I WYPEŁNIANIE OBIEKTÓW NA EKRANIE:

- **SETLINESTYLE (rodzaj, wzór, grubość)** - zmienia rodzaj linii oraz jej grubość

Rodzaje linii to:

- SolidLn = 0 - ciągła
- DottedLn = 1 - kropkowana
- CenterLn = 2 - symetryczna
- DashedLn = 3 - przerywana
- UserBitLn = 4 - o wzorze zdefiniowanym przez użytkownika

Wzór - jest to zdefiniowane przez użytkownika kolejne 16 pikseli nowej nietypowej linii (rodz_lin=4). Jeżeli rodzaj linii jest różny od 4 to pod wzór możemy wstawić jakakolwiek liczbę gdyż nie będzie ona brana pod uwagę.

Rodzaje grubości to:

NormWidth = 1 - linia normalna

ThickWidth = 3 - linia pogrubiona

- **SetFillStyle(wzorzec, kolor)** – określa styl i kolor wypełnianych obiektów, wzór (numer desena 0..11, gdzie 0 – brak wypełnienia), kolor (kolor wypełnienia 0..15)

WZORCE DLA SETFILLSTYLE

EmptyFill	= 0	- wypełnienie kolorem tła
SolidFill	= 1	- wypełnienie ciągłe
LineFill	= 2	- wypełnienie pogrubionymi liniami poziomymi
LtSlashFill	= 3	- wypełnienie liniami pochyłymi w prawo
SlashFill	= 4	- wypełnienie pogrubionymi liniami pochyłymi w prawo
BkSlashFill	= 5	- wypełnienie pogrubionymi liniami pochyłymi w lewo
LtBkSlashFill	= 6	- wypełnienie liniami pochyłymi w lewo
HatchFill	= 7	- wypełnienie siatką pionową
xHatchFill	= 8	- wypełnienie siatką ukośną
InteerleaveFill	= 9	- wypełnienie liniami splecionymi
VideDotFill	= 10	- wypełnienie kropkami
CloseDotFill	= 11	- wypełnienie zagęszczonymi kropkami

PROCEDURA	PROCEDURA WYPEŁNIENIA
RECTANGLE (x1, y1, x2, y2)	BAR (x1, y1, x2, y2)
ELIPSE (x, y, a1, a2, rx, ry)	FILLELIPSE (x, y, rx, ry)
ARC (x, y, a1, a2)	PIESLICE (x, y, a1, a2, r)

FloodFill(x, y, kolor) - procedura wypełnia aktualnym kolorem i wzorem (SetFillStyle) obszar ograniczony brzegiem o kolorze "kolorgranicy" tzn. jeżeli przy wypełnianiu danego obszaru procedura napotka na jakąś np. linię wykonaną kolorem "kolorgranicy" to zachowuje się jakby to był dla niej mur nie do przejścia i zostawia jego i to co się za nim znajduje w spokoju. Punkt x, y jest miejscem od którego procedura zaczyna wypełniać obszar (są to punkty wewnątrz danego obszaru, kolor – kolor konturu)

przykład: setcolor(14);
rectangle(30,30,120,120);
floodfill(50,50,14)

- **GetPixel (x, y)** – odczytuje kolor punktu o współrzędnych (x,y)
przykład: c:=getpixel (x,y)
- **SetColor (k)** – ustalenie koloru rysowanych obiektów na k – kolor (0..15)
- **SetBkColor (kolor)** – ustalenie koloru tła ekranu

*Procedura **SetFillStyle** ustawia styl i kolor wypełnienia. Procedura **FloodFill** wypełnia przestrzeń wypełnieniem ustawionym przez **SetFillStyle**. Zmienna kolor określa kolor ograniczający przestrzeń wypełnianą. Zmienne kolor i styl są zmiennymi typu WORD, a zmienne x, y typu INTEGER.*

RODZAJ PISMA NA EKRANIE:

- **SETTEXTSTYLE** (czcionka; kierunek; wielkość);
 - Czcionka „Font” - rodzaj czcionki, (0..10)
 - Kierunek „Direction” - kierunek (0-poziomo; 1-pionowo)
 - Wielkość „CharSize” - wielkość czcionki (1..10)

przykład: SetColor(4);

SetTextStyle(1,0,4);

OutTextXY(440,550,FREE PASCAL ');

- **OutTextXY** (x, y:integer; s:string) - procedura wyświetla na ekranie tekst określony przez zmienną "s" rozpoczynając od współrzędnych x, y i w aktualnym kolorze (SetColor)

STAŁA	WARTOSC	ZNACZENIE
DefaultFont	0	czcionka rastrowa o wielkości 8x8 (domyślna)
TriplexFont	1	czcionka potrójna
SmallFont	2	czcionka mała
SansSerifFont	3	czcionka bezszeryfowa
GothicFont	4	czcionka gotycka
	5..10	
HorizDir	0	z lewa na prawo
VertDir	1	z dołu do góry
Poziomo z obrotem liter	2	
UsesCharSize	0	definiowana przez użytkownika
GetMaxX textwidth	-----	szerokość tekstu
GetMaxY textheight	-----	wysokość tekstu

KOLORY

nazwa angielska koloru	numer koloru	nazwa polska koloru
BLACK	0	czarny
BLUE	1	niebieski
GREEN	2	zielony
CYAN	3	morski
RED	4	czerwony
MAGENTA	5	fioletowy
BROWN	6	brązowy
LIGHTGRAY	7	jasnoszary
DARKGRAY	8	ciemnoszary
LIGHTBLUE	9	jasnoniebieski
LIGHTGREEN	10	jasnozielony
LIGHTCYAN	11	jasnomorski
LIGHTRED	12	jasnoczerwony
LIGHTMAGENTA	13	jasnofioletowy
YELLOW	14	żółty
WHITE	15	biały

4. Przykłady użycia procedur

Procedura Circle

```
uses graph,crt;
var
  ster,tryb:integer;
begin
  ster:=detect;
  initgraph(ster, tryb, 'C:\TP\BGI' );

  circle(320,240,40);
  repeat until keypressed;
  closegraph;
end.
```

Procedura Rectangle

```
uses graph,crt;
var
  ster,tryb:integer;
begin
  ster:= detect;
  initgraph(ster, tryb, 'C:\TP\BGI' );
  rectangle(1,1,610,450);
  repeat until keypressed;
  closegraph;
end.
```

Procedura FloodFill

```
uses graph,crt;
var
  ster,tryb:integer;
begin
  ster:= detect;
  initgraph(ster, tryb, 'C:\BP\BGI' );
  setcolor(green);
  rectangle(30,30,120,120);
  rectangle(10,10,100,100);
  floodfill(50,50,green);
  repeat until keypressed;
  closegraph;
end.
```

Procedura OutTextXY

```
uses graph,crt;
var
  ster,tryb:integer;
begin
  ster:= detect;
```



```

initgraph(ster, tryb, 'C:\BP\BGI' );
outtextxy(200,100,'GRAFIKA PASCAL');
repeat until keypressed;
closegraph;
end.

```

PROCEDURA CLEARDEVICE

```

uses GRAPH,CRT;
var
  ster,tryb:integer;
begin
  ster:=detect;
  initgraph(ster, tryb, 'C:\TP\BGI' );
  circle(50,50,30);
  delay(2000);
  cleardevice;
  circle(150,200,70);
  repeat until keypressed; {można użyć również procedur: readln; readkey}
  closegraph;
end.

```

ROZWIĄZANIA ZADAŃ NA LEKCJĘ:

Zad 1.

Napisz program rysujący na środku ekranu żółte koło o promieniu 100 wypełnione pogrubionymi liniami pochyłymi w prawo.

```

program zadanie1;
uses graph,crt;
var
  ster,tryb:integer;
begin
  ster:=detect;
  initgraph(ster, tryb, 'C:\TP\BGI' );
  setcolor(14);
  setfillstyle(8,14);
  circle(20,50,10);
  floodfill (20,50,14);
  repeat until keypressed;
  closegraph;
end.

```

Zad. 2.

Napisz program rysujący na całym ekranie monitora flagę Szwajcarii (biały krzyż na czerwonym tle).

```

program zadanie2;
uses crt, graph;
var ster,typ: integer;
begin
  ster:=detect;

```

```

initgraph(ster,typ,'c:\TP\BGI\');
setcolor(4);
rectangle(0,0,319,239);
setfillstyle(1,4);
floodfill(1,1,4);
setfillstyle(solidfill,15);
bar(140,30,190,200);
setfillstyle(solidfill,15);
bar(70,100,250,140);
readln;
closegraph;
end.

```

Zad. 3.

Napisz program rysujący zielony prostokąt wypełniony kropkami otaczający w lewym górnym rogu ekranu.

```

program zadanie3;
uses crt, graph;
var ster,typ: integer;
begin
ster:=detect;
initgraph(ster,typ,'c:\TP\BGI\');
setcolor(2);
rectangle(1,1,100,50);
setfillstyle(10,2);
floodfill(2,5,2);
readln;
closegraph;
end.

```