

МIНIСТЕРСТВО ОСВIТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** 3

з дисципліни “ Основи програмування ”

тема “ Циклічні конструкції ”

|  |  |  |
| --- | --- | --- |
| Виконав(ла)  студент(ка) I курсу  групи КП-83  Клименко Ярослав Олександрович  (*прізвище, ім’я, по батькові*)    варіант № 8 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

Київ 2018

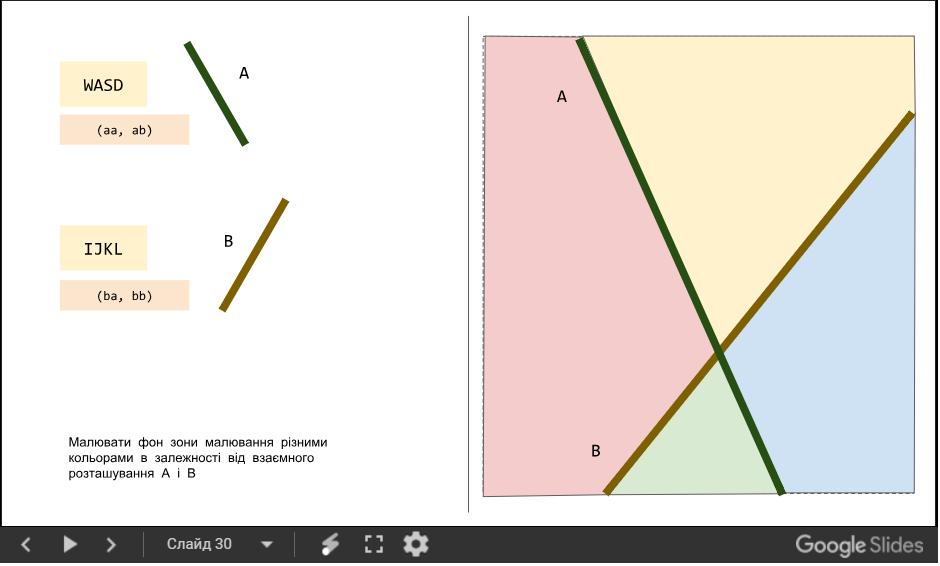
**Мета роботи**

Застосувати на практиці різні види циклічних конструкцій.

Навчитись параметризувати алгоритми для роботи із геометричними об'єктами.

**Постановка завдання**

За допомогою функцій бібліотеки **libprogbase** намалювати у терміналі просту графічну сцену, що складається із об'єктів за варіантом. Запрограмувати можливість керування об'єктами за допомогою вказаних клавіш із перемалюванням сцени.

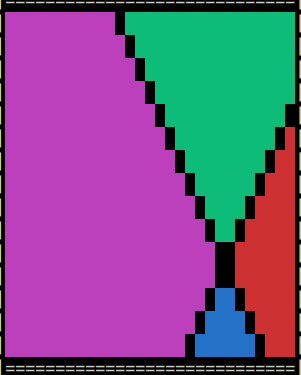
Слайд №30

**Тексти коду програм**

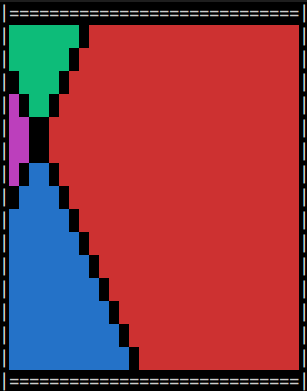
|  |
| --- |
| **main.c** |
| #include <stdio.h>  #include <progbase/console.h>  #include <math.h>  #include <stdlib.h>  void drawLine(int x0, int y0, int x1, int y1, char ch);  int main(void)  {  char key = 0;  // VARS FOR LINE A  int linAx0 = 30;  int linAy0 = -1 \* linAx0 + 34;//4  int linAx1 = 17;  int linAy1 = -1 \* linAx1 + 34;;//17  //VARS FOR LINE B  int linBx0 = 5;  int linBy0 = linBx0 - 5;//0  int linBx1 = 22;  int linBy1 = linBx1 - 5; //17    do {  // FRAME  int zwidth = 0;  int zheight = 0;  const int x0 = 1;  const int width = 30;  const int y0 = 1;  const int height = 16;  Console\_clear();  int ax = 0;  int ay = 0;  const int ax0 = x0 + 1;  const int ay0 = y0 + 1;  // ADD SOME COLORS  Console\_setCursorPosition(ay0, ax0);    for (ay = ay0; ay <= height; ay++)  {  for (ax = ax0; ax <= width; ax++)  {  Console\_setCursorPosition(ay, ax);    if ((ax >= linBx0 + ay) && ax <= linAx0 - ay + linAy0) {  Console\_setCursorAttribute(BG\_GREEN);  } else if (ax >= linAx0 - ay + linAy0 && ax >= linBx0 + ay) {  Console\_setCursorAttribute(BG\_RED);  } else if (ax <= linBx0 + ay && ax <= linAx0 - ay + linAy0){  Console\_setCursorAttribute(BG\_MAGENTA);  } else  Console\_setCursorAttribute(BG\_BLUE);    putchar(' ');  Console\_reset();  }  }  //LINE 1  Console\_setCursorAttribute(BG\_BLACK);  drawLine(linAx0, linAy0, linAx1, linAy1, ' ' );  //LINE 2  drawLine(linBx0, linBy0, linBx1, linBy1, ' ' );    //Console\_reset();    // horizontal lines  for (zwidth = x0; zwidth <= x0 + width; zwidth++)  {  zheight = y0;  Console\_setCursorPosition(zheight, zwidth);  printf("=");  zheight = y0 + height;  Console\_setCursorPosition(zheight, zwidth);  printf("=");  }  // vertical lines  for (zheight = y0; zheight <= y0 + height; zheight++)  {  zwidth = x0;  Console\_setCursorPosition(zheight, zwidth);  printf("|");  zwidth = x0 + width;  Console\_setCursorPosition(zheight, zwidth);  printf("|");  }  //INPUT  Console\_reset();  key = Console\_getChar();    switch(key){  case 'w':  case 'a':  {  if (linAx1 > x0 - 0.4 \* height) {  linAy0 -= 1;  linAx1 -= 1;    }  }  break;    case 's':  case 'd':  {  if (linAy0 < 0.4 \* height ){  linAy0 += 1;  linAx1 += 1;  }  }  break;    case 'k':  case 'j':  {  if (linBx1 > 0.4 \* width) {  linBx0 -= 1;  linBx1 -= 1;  }  }  break;    case 'i':  case 'l':  {  if (linBx1 <= width){  linBx0 += 1;  linBx1 += 1;  }  }  break;    default: ;  }  Console\_reset();  Console\_setCursorPosition(25, 1);    } while (key != ' ');    return 0;  }  void drawLine(int x0, int y0, int x1, int y1, char ch)  {  int dy = abs(y1 - y0);  int dx = abs(x1 - x0);  int sx = x1 >= x0 ? 1 : -1;  int sy = y1 >= y0 ? 1 : -1;  if (x0 > 0 && y0 > 0)  {  Console\_setCursorPosition(y0, x0);  putchar(ch);  }  if (dy <= dx) {  int d = (dy << 1) - dx;  int d1 = dy << 1;  int d2 = (dy - dx) << 1;  for(int x = x0 + sx, y = y0, i = 1; i <= dx; i++, x += sx) {  if ( d >0) {  d += d2;  y += sy;  }  else {  d += d1;  }  // draw point  if (y > 0 && x > 0) {  Console\_setCursorPosition(y, x);  putchar(ch);  }  }  }  else {  int d = (dx << 1) - dy;  int d1 = dx << 1;  int d2 = (dx - dy) << 1;  for(int y = y0 + sy, x = x0, i = 1; i <= dy; i++, y += sy)  {  if ( d >0)  {  d += d2;  x += sx;  }  else  {  d += d1;  }  // draw point  if (x > 0 && y > 0)  {  Console\_setCursorPosition(y, x);  putchar(ch);  }  }  }  } |

**Приклади результатів**

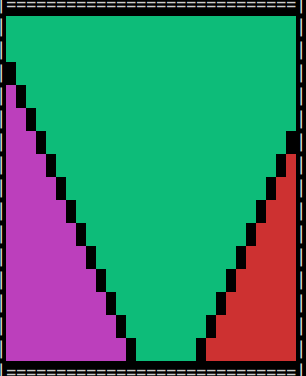
Залежно від положення прямих, робоча зона ділиться на 4 кольори:

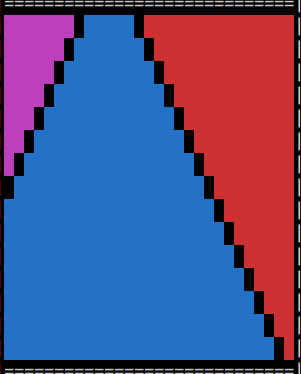


Обидві прямі можуть переміщуватися користувачем, але при цьому поділ зони на кольори не зміниться:



Якщо ж прямі розбивають зону не на 4 сектори, то кількість зафарбованих ділянок зменшується





**Висновки**

Виконуючи дану лабораторну роботу, мені довелося ознайомитися з використанням циклів у мові програмування С, а також бібліотекою libprogbase.

За допомогою циклу з лічильником, мені вдалося проводити побудову різних за складністю фігур та задати їх розмір з допомогою змінних типу integer. Використовуючи цикл з післяумовою, програма може продовжувати свою роботу доти, доки користувач не натисне кнопку виходу ( SPACE). Отже, програма має коректну умову закінчення роботи.

У бібліотеці libprogbase я знайшов велику кількість корисних функцій для роботи з виводом у консоль користувача, які значно спростили написання програми.

Компіляція всього коду відбувалася за допомогою компілятора gcc.