

**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 ”**

|  |  |  |
| --- | --- | --- |
| **Виконала**  **студентка I курсу**  **групи КП-83**  **Снітко Маріанна Дмитрівна**  **(прізвище, ім’я, по батькові)**  **варіант № Slide 4** |  | **Перевірив**  “**\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.**  **викладач**  **Гадиняк Руслан Анатолійович**  **(прізвище, ім’я, по батькові)** |

**Київ 2018**

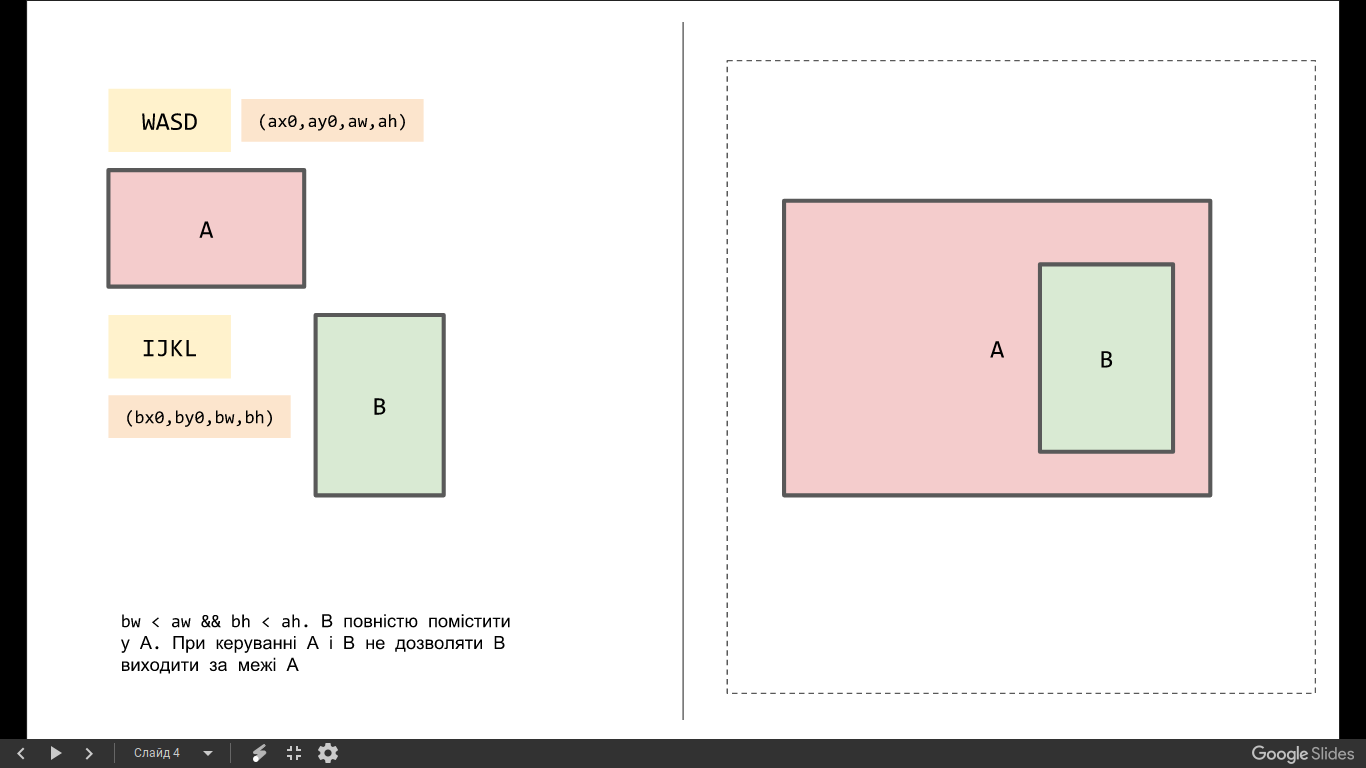
**Мета роботи**

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. Для малювання об'єктів можна використовувати будь-які кольори терміналу. Початкові розміри та позиції об'єктів довільні (можливі обмеження у певних варіантах).

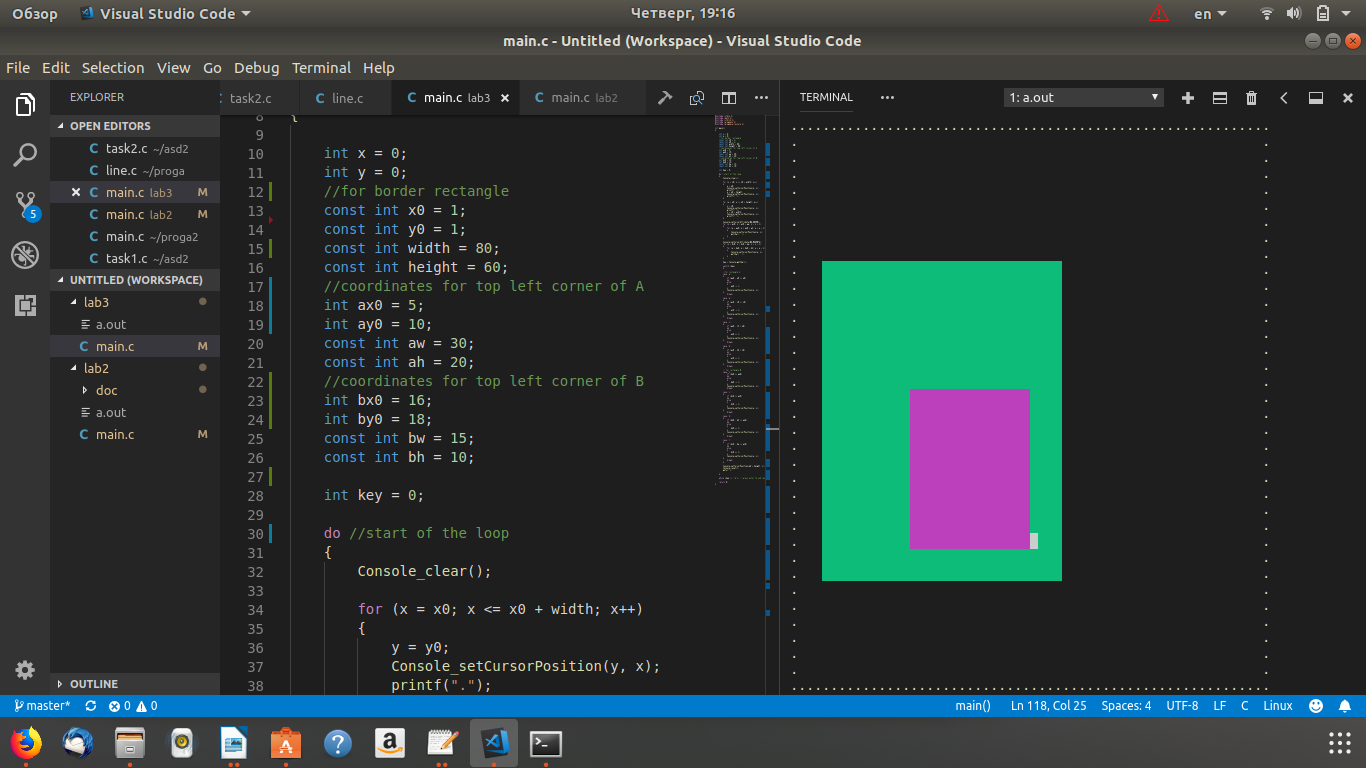
Малювати зону відображення як границі прямокутника довільного розміру та позиції у терміналі. Всі інші об'єкти не повинні при малюванні виходити за границі зони відображення.

****

**Тексти коду програм**

|  |
| --- |
| **main.c** |
| #include <stdio.h>  #include <math.h>  #include <stdlib.h>  #include <progbase.h>  #include <progbase/console.h>  int main()  {  int x = 0;  int y = 0;  //for border rectangle  const int x0 = 1;  const int y0 = 1;  const int width = 60;  const int height = 40;  //coordinates for top left corner of A  int ax0 = 5;  int ay0 = 10;  const int aw = 30;  const int ah = 20;  //coordinates for top left corner of B  int bx0 = 16;  int by0 = 18;  const int bw = 15;  const int bh = 10;  int key = 0;  do //start of the loop  {  Console\_clear();  //drawing horizontal lines  for (x = x0; x <= x0 + width; x++)  {  y = y0;  Console\_setCursorPosition(y, x);  printf(".");  y = y0 + height;  Console\_setCursorPosition(y, x);  printf(".");  }  //drawing vertical lines  for (y = y0; y <= y0 + height; y++)  {  x = x0;  Console\_setCursorPosition(y, x);  printf(".");  x = x0 + width;  Console\_setCursorPosition(y, x);  printf(".");  }  Console\_setCursorAttribute(BG\_GREEN);  for (x = ax0; x < ax0 + aw; x++)  {  for (y = ay0; y < ay0 + ah; y++)  {  Console\_setCursorPosition(y, x);  putchar(' ');  }  }  Console\_setCursorAttribute(BG\_MAGENTA);  for (x = bx0; x < bx0 + bw; x++)  {  for (y = by0; y < by0 + bh; y++)  {  Console\_setCursorPosition(y, x);  putchar(' ');  }  }  key = Console\_getChar();  switch (key)  {  //for rectangle A    case 'w':  {  if (ay0 <= 2)  {}  else  {  ay0 -= 1;  by0 -= 1;  }  Console\_setCursorPosition(y, x);  }  break;  case 'a':  {  if (ax0 <= 2)  {}  else  {  ax0 -= 1;  bx0 -= 1;  }  Console\_setCursorPosition(y, x);  }  break;  case 's':  {  if (ay0 + ah == y0 + height)  {}  else  {  ay0 += 1;  by0 += 1;  }  Console\_setCursorPosition(y, x);  }  break;  case 'd':  {  if (ax0 + aw == x0 + width)  {}  else  {  ax0 += 1;  bx0 += 1;  }  Console\_setCursorPosition(y, x);  }  break;  // for rectangle B  case 'i':  {  if (by0 <= ay0)  {}  else  {  by0 -= 1;  }  Console\_setCursorPosition(y, x);  }  break;  case 'j':  {  if (bx0 <= ax0)  {}  else  {  bx0 -= 1;  }  Console\_setCursorPosition(y, x);  }  break;  case 'k':  {  if (by0 - bh >= ay0)  {}  else  {  by0 += 1;  }  Console\_setCursorPosition(y, x);  }  break;  case 'l':  {  if (bx0 - bw >= ax0)  {}  else  {  bx0 += 1;  }  Console\_setCursorPosition(y, x);  }  break;  }  Console\_setCursorPosition(y0 + height, 1);  Console\_reset();  puts("");  }  while (key != '\n'); // press enter to end loop  return 0;  } |

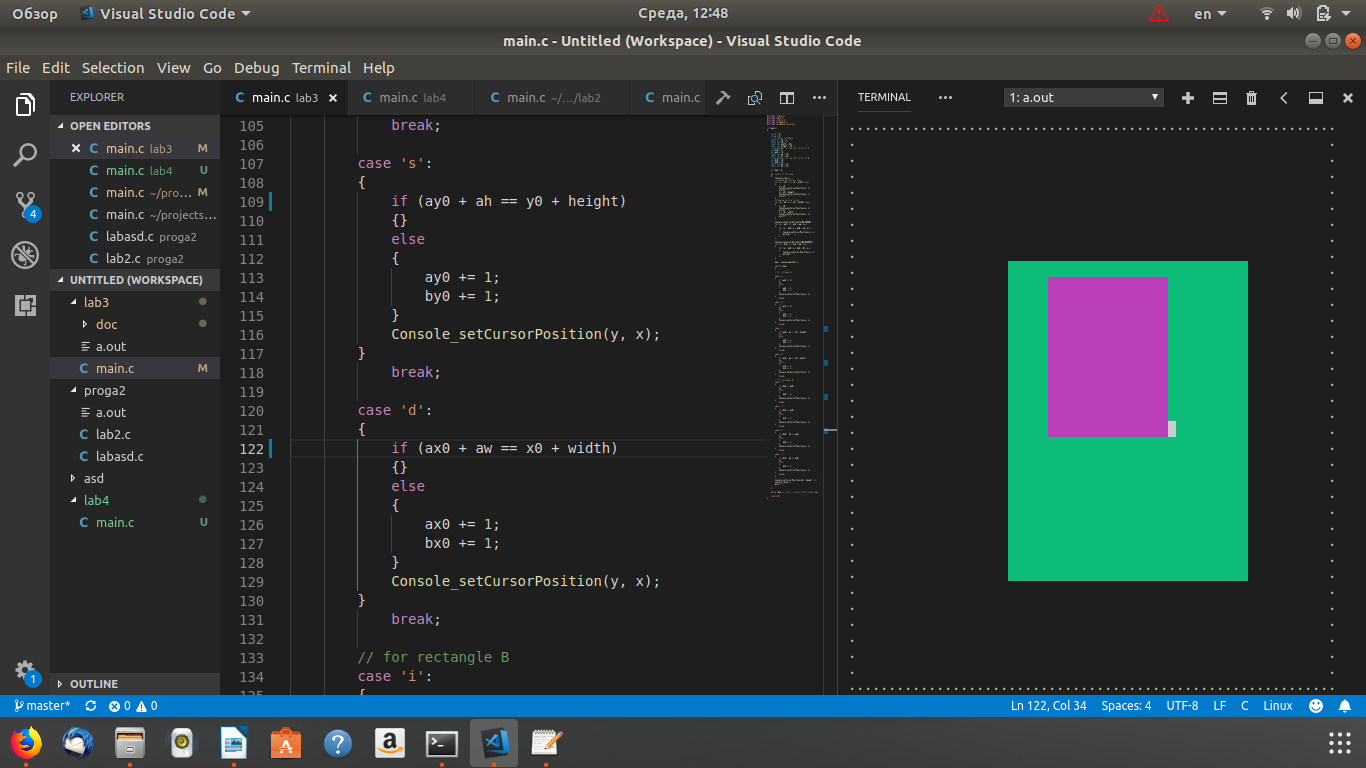
**Приклади результатів**

****

Початкова позиція графічної сцени:

Прямокутник В повністю поміщений у прямокутник А. Також при керуванні В не виходить за межі А, а А не виходить за зону відображення.

Графічна сцена після руху обʼєктів:



**Висновки**

Протягом виконання цієї лабораторної роботи я ознайомилася із програмуванням на мові С у спеціалізованому редакторі коду.

Більше того, на практиці було застосовано різні види циклічних конструкцій. Також я навчилася параметризувати алгоритми для роботи із геометричними об'єктами.

Для виконання завдання були використані на практиці цикли do while та цикл з лічильником for.

Компіляція всього коду відбувалася за допомогою компілятора gcc.