

**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 № 6**

**з дисципліни “ Основи програмування ”**

**тема “ Структури даних ”**

|  |  |  |
| --- | --- | --- |
| **Виконала**  **студентка I курсу**  **групи КП-83**  **Снітко Маріанна Дмитрівна**  **(прізвище, ім’я, по батькові)**  **варіант № 14** |  | **Перевірив**  “**\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.**  **викладач**  **Гадиняк Руслан Анатолійович**  **(прізвище, ім’я, по батькові)** |

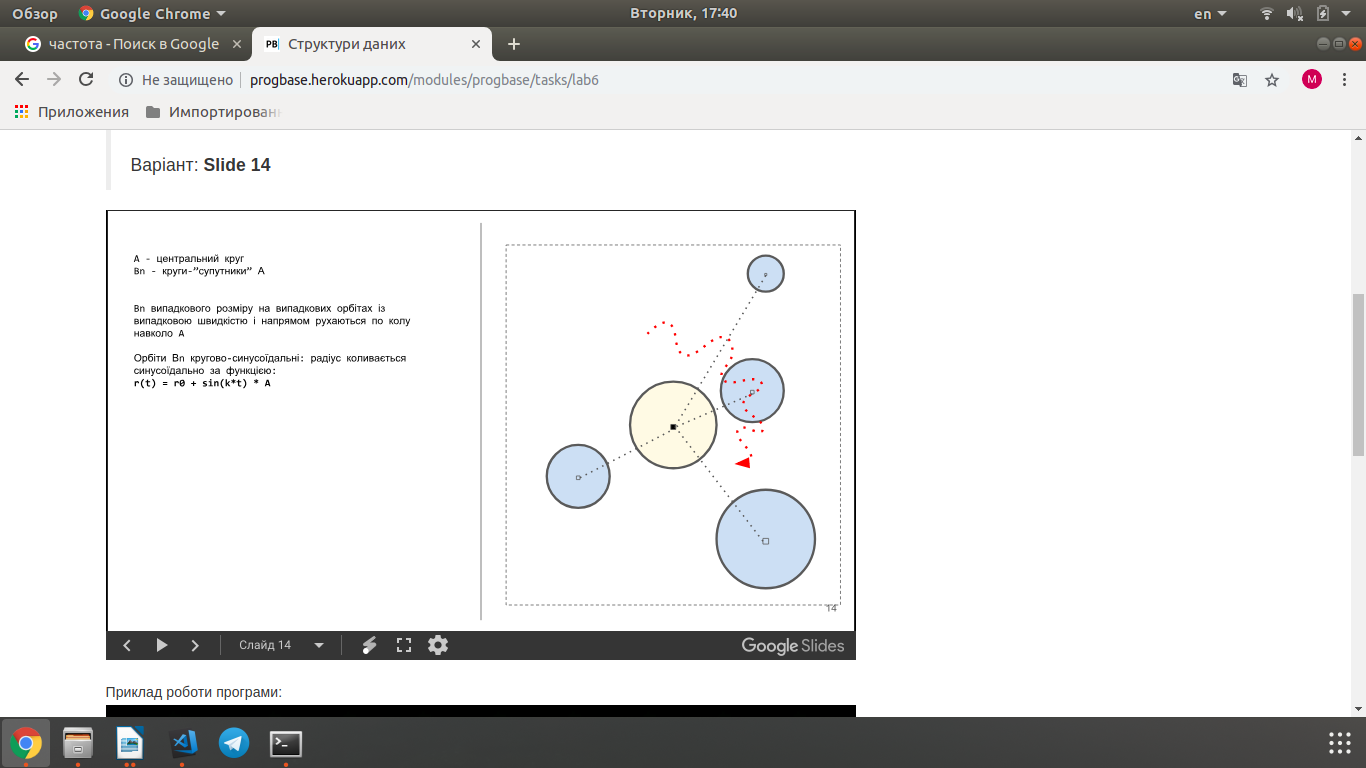
**Київ 2018**

**Мета роботи**

Навчитися працювати зі структурами даних.

**Постановка завдання**

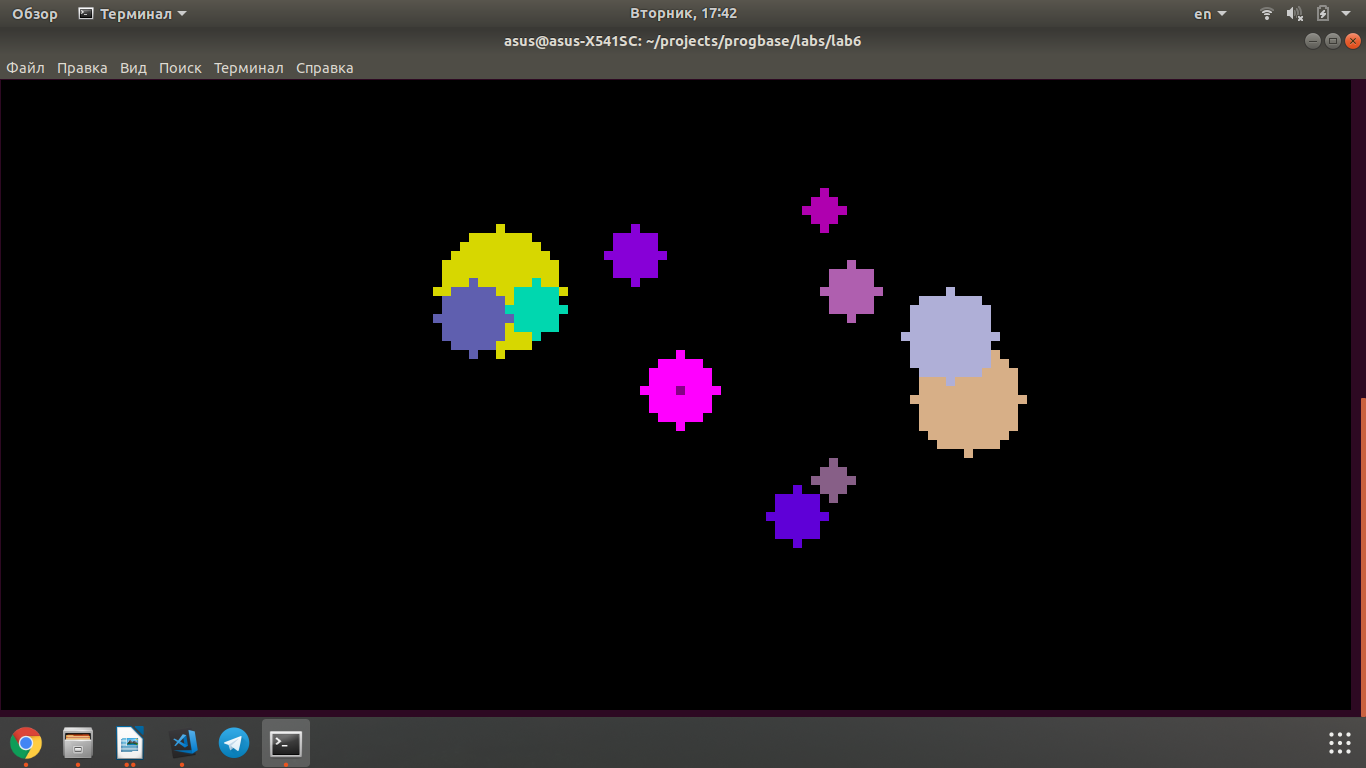
За допомогою функцій модуля **Canvas** бібліотеки **libprogbase** (progbase/canvas.h, версія бібліотеки не менше [0.6.0](https://github.com/PublicHadyniak/libprogbase/releases/tag/0.6.0)) намалювати у терміналі графічну сцену, що складається із графічних об'єктів та піддається певним правилам взаємодії за варіантом.Виділити відповідні типи даних графічних об'єктів як структури і використовувати змінні цих структур для зберігання їх параметрів. Однотипні змінні об'єднати у масиви структур.

****

**Тексти коду програм**

|  |
| --- |
| **main.c** |
| // gcc main.c -lprogbase -lm  #include <stdlib.h>  #include <stdio.h>  #include <math.h>  // libprogbase  #include <progbase.h>  #include <progbase/console.h>  #include <progbase/canvas.h>  #include <time.h>  struct Vec2D  {  float x;  float y;  };  struct Color  {  int red;  int green;  int blue;  };  struct ball  {  struct Vec2D loc;  float radius;  float sinRadius;  struct Color fillColor;  float radAmplitude;  float radFrequency;  float rotRadius;  float rotAlpha;  float rotSpeed;  };  int main()  {  float pi = 3.14159;  srand(time(0));  Console\_clear();  struct ConsoleSize consoleSize = Console\_size();  int width = consoleSize.columns;  int height = consoleSize.rows \* 2;  Canvas\_setSize(width, height);  Canvas\_invertYOrientation();  const int nBalls = 10;  struct ball balls[nBalls];  struct Vec2D center = {width / 2, height / 2};  const int delay = 30;  float dt = delay / 1000.0;  float t = 0;  for (int i = 0; i < nBalls; i++)  {  balls[i].radius = rand() % (7 - 2 + 1) + 2;  balls[i].fillColor.red = rand() % (255 - 0 + 1) + 0;  balls[i].fillColor.green = rand() % (255 - 0 + 1) + 0;  balls[i].fillColor.blue = rand() % (255 - 0 + 1) + 0;  balls[i].rotRadius = rand() % (36 - 15 + 1) + 15;  balls[i].rotAlpha = (rand() / (float)RAND\_MAX) \* 2 \* pi + (-pi);  balls[i].rotSpeed = (rand() / (float)RAND\_MAX) \* 2 \* pi + (-pi);  balls[i].radFrequency = rand() % (25 - 15 + 1) + 15;  balls[i].radAmplitude = rand() % (5 - 2 + 1) + 2;  }  while (1)  {  for (int i = 0; i < nBalls; i++)  {  t += dt;  balls[i].loc.x = center.x + cos(balls[i].rotAlpha) \* balls[i].sinRadius;  balls[i].loc.y = center.y + sin(balls[i].rotAlpha) \* balls[i].sinRadius;  balls[i].rotAlpha += balls[i].rotSpeed \* dt;  balls[i].sinRadius = balls[i].rotRadius + sin(balls[i].radFrequency \* t) \* balls[i].radAmplitude;  }  Canvas\_beginDraw();  Canvas\_setColorRGB(255, 0, 255);  Canvas\_fillCircle((width / 2), (height / 2), 4);  Canvas\_setColorRGB(125, 5, 125);  Canvas\_putPixel((width / 2), (height / 2));  for (int i = 0; i < nBalls; i++)  {  struct ball ball1 = balls[i];  Canvas\_setColorRGB(ball1.fillColor.red, ball1.fillColor.green, ball1.fillColor.blue);  Canvas\_fillCircle(ball1.loc.x, ball1.loc.y, ball1.radius);  }  Canvas\_endDraw();  sleepMillis(delay);  }  return 0;  } |

**Приклади результатів**

****

**Висновок**

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

Більше того, під час виконання лабораторної роботи я навчилася працювати зі структурами даних

Компіляція всього коду відбувалася за допомогою компілятора gcc.

Отже, мета лабораторної роботи досягнена, всі завдання виконані і їх розвʼязання наведені.