

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

ІМЕНІ ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики

Кафедра програмного забезпечення комп’ютерних систем

**Лабораторна робота №** **4**

з дисципліни “Основи програмування”

тема “Структури даних”

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Виконав  студент I курсу | | групи КП-01 | | Пецеля Артем Володимирович | | (*прізвище, ім’я, по батькові*) |   варіант №12 |  | Перевірив  “\_\_\_\_” “\_\_\_\_\_\_\_\_\_\_\_\_” 20\_\_\_ р.  викладач  Гадиняк Руслан Анатолійович  (*прізвище, ім’я, по батькові*) |

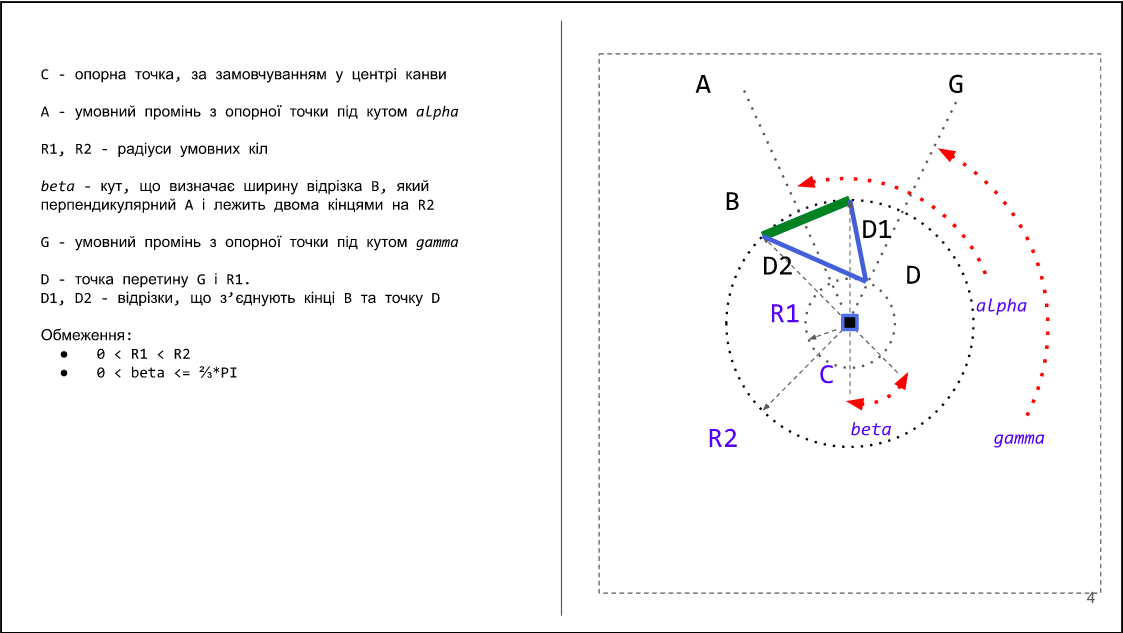
Київ 2020

**Мета роботи**

1. Навчитися створювати нові типи даних структур та використовувати змінні цих типів.
2. Навчитись виконувати операції над значеннями структур даних та використовувати значення-структури у користувацьких функціях. Навчитися використовувати масиви структур даних, посилання на структури та масиви структур даних у користувацьких функціях.

**Постановка завдання**

Намалювати у консолі набір графічних взаємозалежних об'єктів, параметри яких можна змінювати вводом користувача.



Графічні об'єкти, імена яких виділені на слайді варіанту синім кольором можна змінювати командами користувача. Графічні об'єкти з чорними назвами залежать від інших графічних об'єктів і їх параметри потрібно автоматично обчислювати на основі параметрів інших об'єктів.

Варіант завдання містить обмеження деяких параметрів. При зміні параметрів не допускати вихід їх значень за допустимі межі.

**Аналіз вимог і проектування**

Для виконання лабораторної роботи було застосовано три користувацькі структури даних: Point, Circle, Ray.

1. Point - точка на декартовій площині. Має цілочисельну координати x та y.
2. Circle - коло на декартовій площині. Має цілочисельний радіус radius та точку центра center.
3. Ray - промінь, що задається через кут та початкову точку. Дробовий кут - angle, точка початку відкладання - start.

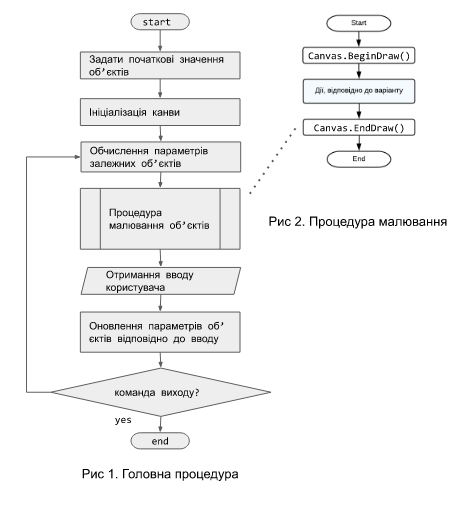
Для координат та відстаней було використано цілочисельні значення, оскільки завданням передбачено вивід в консоль, що має координатку решітку з цілих чисел.

Для знаходження похідних графічних об’єктів була застосовано дві формули пошуку точок перетину кола з променем - окремо для координат x та y:

,

де x та y - шукані координати, circle.center.x та circle.center.y - координати центра кола, circle.radius - радіус кола, ray.angle - кут, під яким відкладений промінь.

Алгоритм роботи програми:



**Текст коду програми**

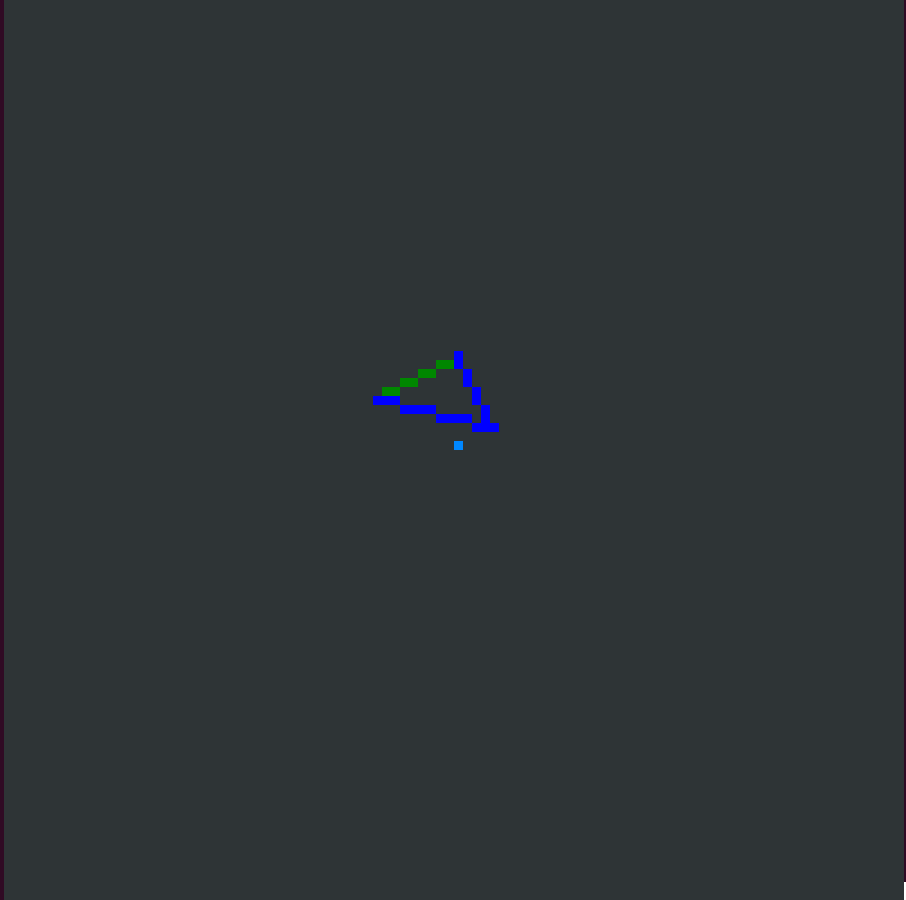
|  |
| --- |
| **Program.cs** |
| using System;  using static System.Math;  using static System.Console;  using Progbase.Procedural;  namespace lab4  {  class Program  {  struct Point  {  public int x;  public int y;  }  struct Circle  {  public Point center;  public int radius;  }  struct Ray  {  public double angle;  public Point start;  }  static void Main(string[] args)  {  const int size = 100;  Clear();  int cx = size/2;  int cy = size/2;    int R1 = 5;  int R2 = 10;    int alpha = 120;  int beta = 60;  int gamma = 30;  Clear();  Canvas.SetOrigin(1,1);  Canvas.SetSize(size, size);  Canvas.InvertYOrientation();  bool exit = false;  do {  Point C = new Point {x = cx, y = cy};  Circle circle1 = new Circle {center = C, radius = R1};  Circle circle2 = new Circle {center = C, radius = R2};  Ray A = new Ray {angle = ConvertDegToRad(alpha), start = C};  Ray G = new Ray {angle = ConvertDegToRad(gamma), start = C};  Point D = new Point {x = (int)(circle1.center.x + circle1.radius \* Cos(G.angle)), y = (int)(circle1.center.y + circle1.radius \* Sin(G.angle))};  Ray A1 = new Ray {angle = A.angle - ConvertDegToRad(beta)/2.0, start = A.start};  Ray A2 = new Ray {angle = A.angle + ConvertDegToRad(beta)/2.0, start = A.start};  Point D1 = new Point {x = (int)(circle2.center.x + circle2.radius \* Cos(A1.angle)), y = (int)(circle2.center.y + circle2.radius \* Sin(A1.angle))};  Point D2 = new Point {x = (int)(circle2.center.x + circle2.radius \* Cos(A2.angle)), y = (int)(circle2.center.y + circle2.radius \* Sin(A2.angle))};  DrawObjects(C, D, D1, D2, circle1, circle2);    ConsoleKeyInfo keyInfo = ReadKey();  if (keyInfo.Key == ConsoleKey.W && circle2.radius + cy < size) cy++;  else if (keyInfo.Key == ConsoleKey.S && cy - circle2.radius > 0) cy--;  else if (keyInfo.Key == ConsoleKey.A && cx - circle2.radius > 0) cx--;  else if (keyInfo.Key == ConsoleKey.D && circle2.radius + cx < size) cx++;  else if (keyInfo.Key == ConsoleKey.UpArrow) alpha++;  else if (keyInfo.Key == ConsoleKey.DownArrow) alpha--;  else if (keyInfo.Key == ConsoleKey.LeftArrow) gamma++;  else if (keyInfo.Key == ConsoleKey.RightArrow) gamma--;  else if (keyInfo.Key == ConsoleKey.T && beta > 0) beta--;  else if (keyInfo.Key == ConsoleKey.Y && beta <= 120) beta++;  else if (keyInfo.Key == ConsoleKey.G && R1 > 0) R1--;  else if (keyInfo.Key == ConsoleKey.H && R1 < R2) R1++;  else if (keyInfo.Key == ConsoleKey.B && R2 > R1) R2--;  else if (keyInfo.Key == ConsoleKey.N && cy + R2 + 1 < size && cx + R2 + 1 < size && cy - R2 - 1 > 0 && cx - R2 - 1 > 0) R2++;  else if (keyInfo.Key == ConsoleKey.Escape) exit = true;  } while (exit == false);  }  static void DrawObjects(Point C, Point D, Point D1, Point D2, Circle circle1, Circle circle2)  {  Canvas.BeginDraw();  Canvas.SetColor("#1E90FF");  Canvas.PutPixel(C.x, C.y);  Canvas.SetColor("#008000");  Canvas.StrokeLine(D1.x, D1.y, D2.x, D2.y);  Canvas.SetColor("#0000FF");  Canvas.StrokeLine(D1.x, D1.y, D.x, D.y);  Canvas.StrokeLine(D2.x, D2.y, D.x, D.y);  Canvas.EndDraw();  }  static double ConvertDegToRad(int degree)  {  double rad = PI \* degree / 180.0;  return rad;  }  }  } |

**Приклади результатів**

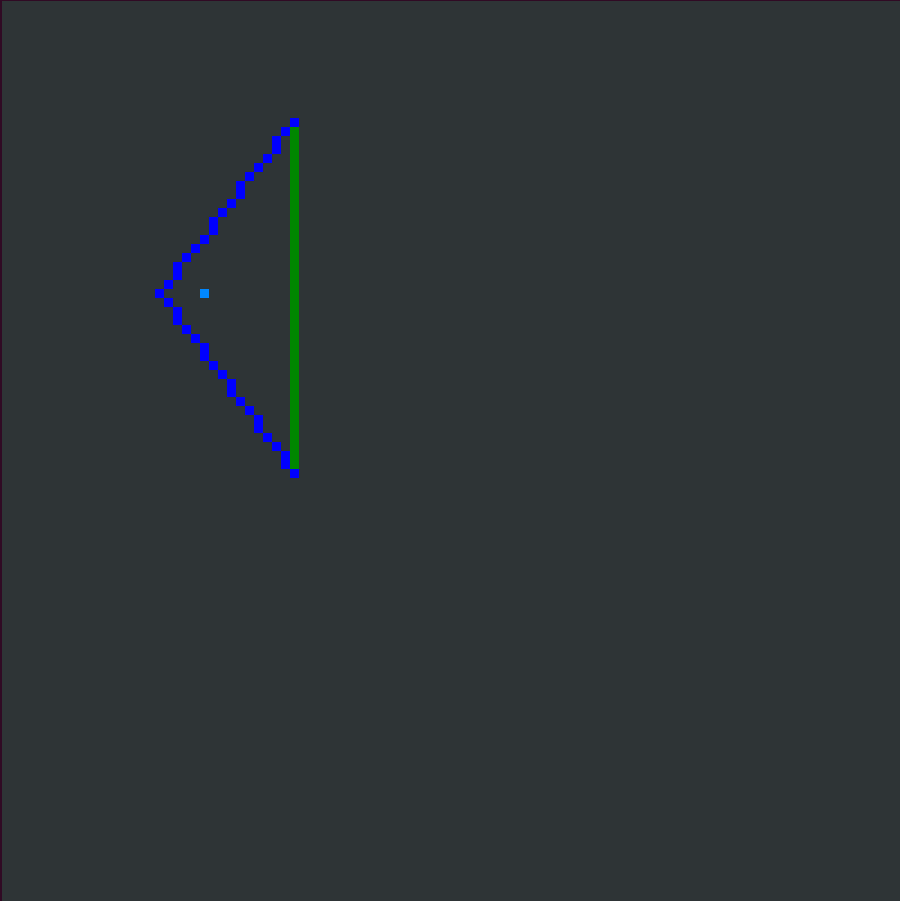
Клавіші, якими користувач може змінювати певні параметри:

1. W - зсув точки C вгору
2. S - зсув точки С вниз
3. A - зсув точки С вліво
4. D - зсув точки С вправо
5. ↑ - збільшення кута alpha
6. ↓ - зменшення кута alpha
7. ← - зменшення кута gamma
8. → - збільшення кута gamma
9. T - зменшення кута beta
10. Y - збільшення кута beta
11. G - зменшення радіуса R1
12. H - збільшення радіуса R1
13. B - зменшення радіуса R2
14. N - збільшення радіуса R2

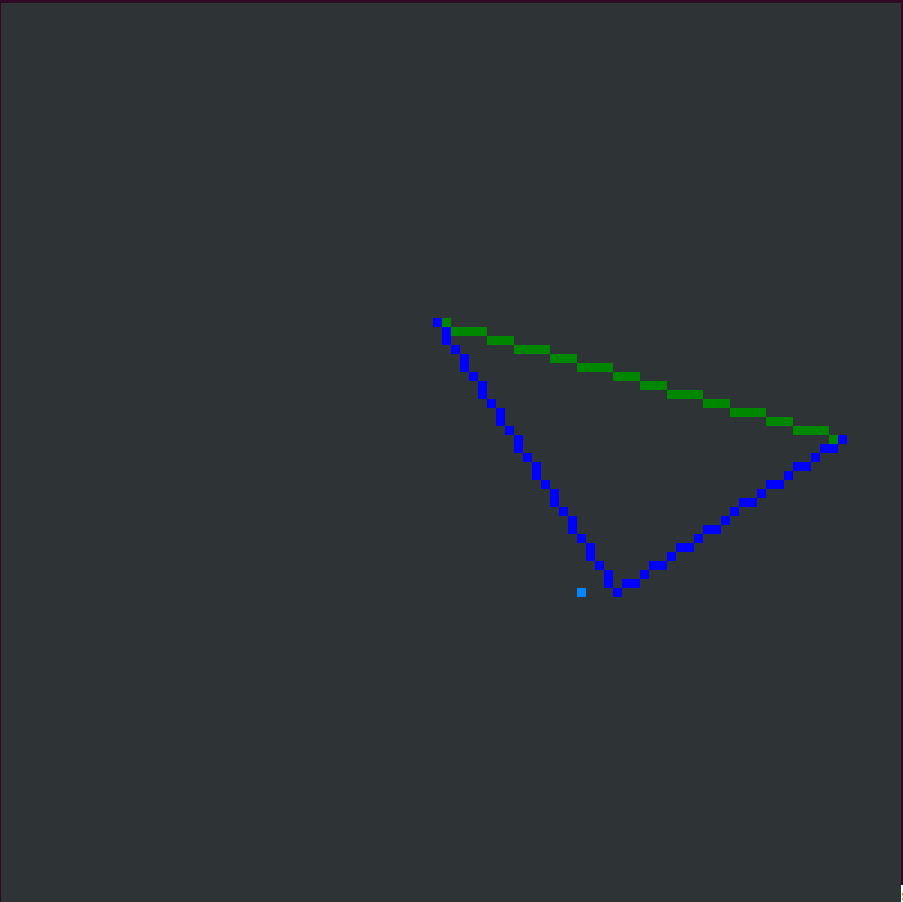
**Приклад 1:**

****

**Приклад 2:**

****

**Приклад 3:**

****

**Висновки**

Було створено різні структури даних та застосовано їх змінні для створення графічних об’єктів. Також на основі даних, взятих з вже існуючої структури, після деяких обрахунків було створено нові структури, що містили в собі деякі інші графічні об’єкти.

Побудова графічних об’єктів відбувалася за допомогою пакету Progbase.Canvas.

Компіляція коду відбувалася за допомогою утиліти dotnet.