|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ *Робототехника и комплексная автоматизация*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

по дисциплине: «Объектно-ориентированное программирование»

|  |  |  |
| --- | --- | --- |
| Студент |  | Камалов Антон Павлович |
| Группа |  | РК6-25Б |
| Тип задания |  | Лабораторная работа №1 |
| Тема лабораторной работы |  | Кодировки, Вариант Р16 |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \_**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_ \_\_**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2023 г.*

# Вариант задания

# Разработать объектно-ориентированную программу для вычисления декартовых координат вершин правильного восьмиугольника с заданной длиной стороны, центр которого смещен на юго-восток от начала координат, а две диагонали параллельны осям координат. Длина стороны и смещение центра восьмиугольника должна передаваться программе аргументами командной строки ее вызова, а полученные координаты его вершин должны отображаться в потоке стандартного вывода. Программная реализация вычислений должна быть основана на разработке класса точки с приватными полями для ее декартовых координат, публичными методами доступа к ним и конструктором инициализации их значений. Кроме того, в этом классе нужно предусмотреть статический метод для конструирования точки по ее полярным координатам.

# Описание решения

1. В функции main создаются 6 переменных типа float: длина стороны правильного восьмиугольника, угол, с помощью которого будем находить вершины, число Пи, радиус описанной окружности, смещения по х и у относительно начала координат; 1 переменную-счётчик типа int.
2. Переменным, отвечающим за смещение или сторону треугольника, присваиваем соответствующие значения из аргументов командной строки. Если количество аргументов меньше 3, то функция main возвращает значение -1, происходит завершение работы программы.
3. Если смещение указано неверно, то функция main возвращает значение -2, происходит завершение работы программы.
4. С помощью цикла while, воспользовавшись статическим методом класса Polar, создаём вершины правильного восьмиугольника, меняя угол.
5. С помощью цикла do-while выводим получившиеся координаты вершин.

# Текст программы

|  |  |
| --- | --- |
| 01:  02:  03:  04:  05:  06:  07:  08:  09:  10:  11:  12:  13:  14:  15:  16:  17:  18:  19:  20:  21:  22:  23:  24:  25:  26:  27:  28:  29:  30:  31:  32:  33:  34:  35:  36:  37:  38:  39:  40:  41:  42:  43:  44:  45:  46:  47:  48:  48:  49:  50:  51:  52:  53:  54:  55:  56: | class Point {  private:  float x, y;  Point (float \_x, float \_y): x(\_x),y(\_y) {};  public:  Point(): x(0), y(0) {};  float getX() {return x;}  float getY() {return y;}  // Статический метод класса  static Point polar(float, float);  };  // Перевод координат точки из полярной в декартову систему  Point Point::polar(float radius, float f) {  Point p(radius\*cos(f), radius\*sin(f));  return p;  };  int main(int argc, char\*\* argv) {  float side = 0.0;  float angle = 0.0;  float pi = acos(-1.0);  float radius;  float deltaX = 1.0, deltaY = -1.0;  Point vertex[8];  int i = 0;  // Проверка на количество аргументов  if (argc > 2) {  side = atof(argv[1]);  deltaX = atof(argv[2]);  deltaY = atof(argv[3]);  } else{  std::cout << "Invalid count of arguments\n";  return -1;  }  radius = side\*sqrt(4+2\*sqrt(2))/2;  // Проверка на расположение точки  if (deltaX <= 0 || deltaY >= 0){  std::cout << "Invalid coordinates of point";  return -2;  }  while (i < 8){  // Создание вершины восьмиугольника  vertex[i] = Point::polar(radius,angle);  angle+=(pi/4.0);  i++;  }  do{  // Вывод координат вершин восьмиугольника  printf("Point %d: %f ; %f\n", i, vertex[i].getX()+deltaX, vertex[i].getY()+deltaY);  i--;  }while(i>0);  return 0;  } |

**Результаты работы программы**

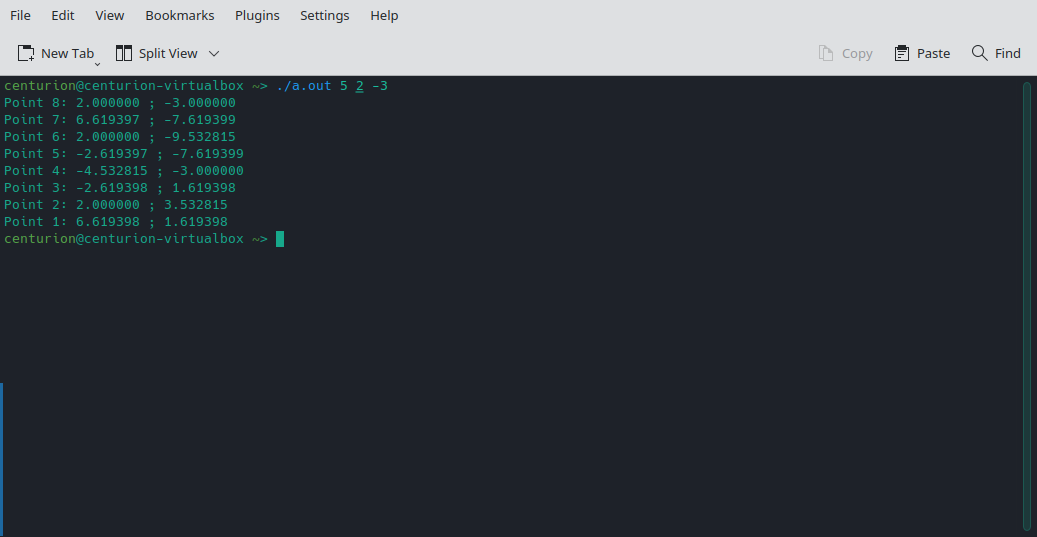


Рисунок 1. Выполнение программы без

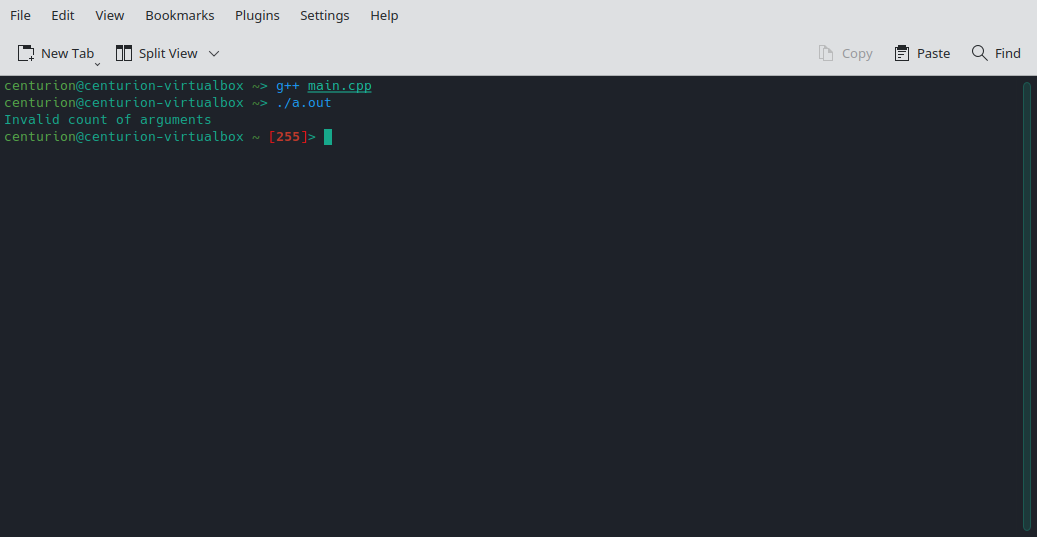


Рисунок 2. Выполнение программы с недостаточным количеством аргументов

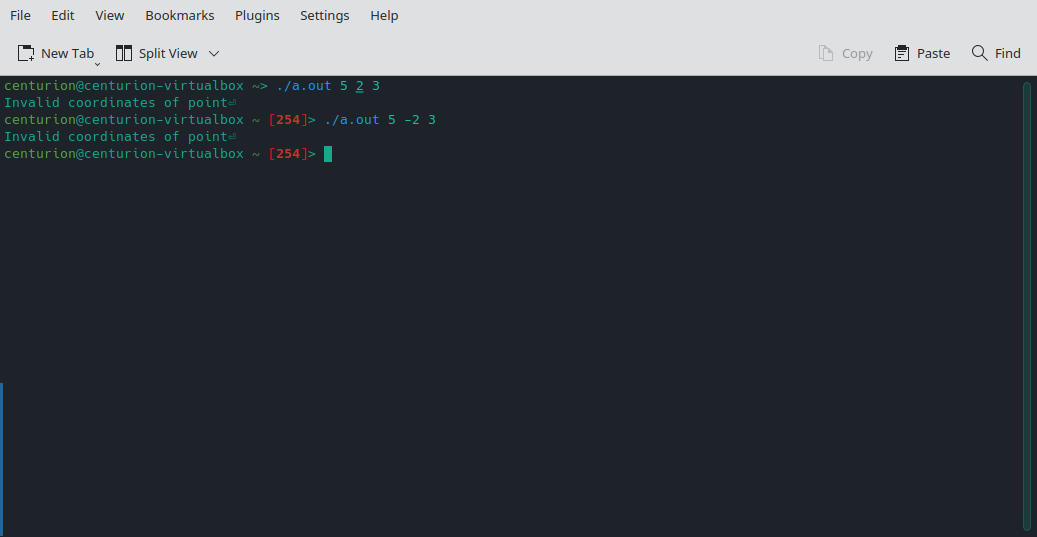


Рисунок 3. Выполнение программы с неправильным расположением центра правильного восьмиугольника