

**Московский авиационный институт
(Национальный исследовательский университет)**

Факультет: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Объектно-ориентированное программирование»

Лабораторная работа № 4

Тема: Метапрограммирование C++

Студент: Пермяков Никита
Александрович

Группа: 80-208

Преподаватель: Чернышов Л.Н.

Дата:

Оценка:

Москва, 2020

Постановка задачи

Разработать шаблоны классов согласно варианту задания. Параметром шаблона должен являться скалярный тип данных задающий тип данных для оси координат. Классы должны иметь только публичные поля. В классах не должно быть методов, только поля. Фигуры являются фигурами вращения (равнобедренными), за исключением трапеции и прямоугольника. Для хранения координат фигур необходимо использовать шаблон `std::pair`.

1. Функция `print` печати фигур на экран `std::cout` (печататься должны координаты вершин фигур). Функция должна принимать на вход `std::tuple` с фигурами, согласно варианту задания (минимум по одной каждого класса).
2. Функция `square` вычисления суммарной площади фигур. Функция должна принимать на вход `std::tuple` с фигурами, согласно варианту задания (минимум по одной каждого класса).

Создать программу, которая позволяет:

- Создает набор фигур согласно варианту задания (как минимум по одной фигуре каждого типа с координатами типа `int` и координатоми типа `double`).
- Сохраняет фигуры в `std::tuple`
- Печатает на экран содержимое `std::tuple` с помощью шаблонной функции `print`.
- Вычисляет суммарную площадь фигур в `std::tuple` и выводит значение на экран.

При реализации шаблонных функций допускается использование вспомогательных шаблонов `std::enable_if`, `std::tuple_size`, `std::is_same`.

Вариант: 25

- Треугольник
- Квадрат
- Прямоугольник

Цель:

- Изучение основ работы с шаблонами (`template`) в C++;
- Изучение шаблонов `std::pair`, `std::tuple`

- Получение навыка работы со специализацией шаблонов и идиомой SFINAE

Описание программы

1. Программа выполняет определённые действия по введённым командам:
 - a. 0 — выход из программы;
 - b. 1,2,3 — создание фигуры (Квадрат, Прямоугольник, Восьмиугольник соответственно), получение вершин через ввод, проверка, вывод данных вершин, вычисление центра и площади;
 - c. 4 — создание кортежа как производного четырёхугольника, получение вершин через ввод, проверка, вывод данных вершин, вычисление центра и площади;
2. Шаблонная функция `print()` печатает координаты всех точек данной фигуры или кортежа. Она определена для моих фигур и `tuple`. Во втором случае все дело вычисляется рекурсивно.
3. Функция `center()` возвращает точку с x — деление суммы x координат всех точек данной фигуры на их количество, y — аналогично x . Она определена для моих фигур и `tuple`. Во втором случае все дело вычисляется рекурсивно;
4. Функция `area()` вычисляет площадь данной фигуры или совокупности точек в кортеже в зависимости от типа фигуры по методу Гаусса (формула землемера, метод шунтирования) и возвращает это значение.

Набор тестов

Пояснение:

- 1) На ввод подается число либо строка выбора действия из текстового интерфейса
- 2) В случае ввода строки — принимается еще одна строка, указывающая на фигуру, с которой производить действие
- 3) При необходимости указывается ID фигуры

- 4) Выводится информация о состоянии вектора фигур, происходит проверка тестов

Test 1

Test 2

Результаты выполнения тестов

Листинг программы

CMakeLists.txt

```
cmake_minimum_required(VERSION 3.5)
```

```
project(lab3)
```

```
add_executable(lab3
```

```
    main.cpp
```

```
    figure.cpp
```

```
    myvector.cpp
```

```
    point.cpp
```

```
    octagon.cpp
```

```
    square.cpp
```

```
    triangle.cpp
```

```
)
```

```
set_property(TARGET lab3 PROPERTY CXX_STANDARD 11)
```

```
set(CMAKE_CXX_FLAGS "${CMAKE_CXX_FLAGS} -Wall -Wextra -g")
```

Ссылка на репозиторий

https://github.com/nikit34/oop_exercise_03

Объяснение результатов работы программы

- 1) Пользователю предоставляется 4 опции: задать фигуру (квадрат, треугольник или восьмиугольник), вывести информацию для каждой фигуры (координаты точек, площадь и геометрический центр), вывести общую площадь всех фигур и удалить фигуру по индексу.
- 2) Перед занесением фигур в вектор каждая фигура проверяется. У квадрата проверяется перпендикулярность и равенство сторон, у треугольника — сумма двух сторон не может быть больше третьей стороны. После чего указатель на созданную фигуру заносится в вектор `figures`.
- 3) Вывод информации о всех фигурах производится с помощью цикла. Поочередно перебираются все элементы вектора `figures`, и с помощью метода `Print()` выводятся координаты, площадь и геометрический центр каждой из фигур.
- 4) Общая площадь фигур находится посредством суммирования результата работы метода `Area()` для всех фигур вектора.
- 5) Если пользователь вводит «0», то считывание завершается, а все фигуры удаляются из памяти с помощью `delete`.

Вывод

В ходе работы были приобретены навыки работы с шаблонами и кортежами в C++. Написана программа, производящая операции с помощью шаблонов и работающая с кортежами. Создал базовый класс и 3 производных от него класса, которые посредством override методов переопределяли виртуальные методы базового класса.

Список литературы

1. Перегрузка операторов C++ [Электронный ресурс]. URL:

<https://metanit.com/cpp/tutorial/5.14.php>

(дата обращения: 29.09.2020).

2. Битовые операции C++ [Электронный ресурс]. URL:

<http://www.c-cpp.ru/books/bitovye-operator>

(дата обращения: 29.09.2020).