

Лабораторная работа № 04

Тема: Основы метапрограммирования

Цель:

- Изучение основ работы с шаблонами (template) в C++;
- Изучение шаблонов умных указателей

Порядок выполнения работы

1. Ознакомиться с теоретическим материалом.
2. Получить у преподавателя вариант задания.
3. Реализовать задание своего варианта в соответствии с поставленными требованиями.
4. Подготовить тестовые наборы данных.
5. Создать репозиторий на GitHub.
6. Отправить файлы лабораторной работы в репозиторий.
7. Отчитаться по выполненной работе путём демонстрации работающей программы на тестовых наборах данных (как подготовленных самостоятельно, так и предложенных преподавателем) и ответов на вопросы преподавателя (как из числа контрольных, так и по реализации программы).

Требования к программе

Разработать шаблоны классов согласно варианту задания. Параметром шаблона должен являться скалярный тип данных задающий тип данных для оси координат. Фигуры являются фигурами вращения (равнобедренными), за исключением трапеции и прямоугольника (эти фигуры просто должны быть вписаны в круг). Для хранения координат фигур необходимо реализовать свой шаблон **template <class T> Point**, в качестве параметра шаблона должен быть тип для переменных координат. Проверку на то что тип T скалярный реализовать с помощью **concept**.

Разработать классы согласно варианту задания, классы должны наследоваться от базового шаблонного класса **Figure<T>**, где T скалярный тип подставляющийся в Point. При этом сами точки должны хранится в виде умных указателей **std::unique_ptr<Point<T>>** Все классы должны поддерживать набор общих методов из предыдущей лабораторной работы.

Создать программу, которая позволяет:

- Запрещается использовать сырые указатели
- Вводить из стандартного ввода **std::cin** фигуры, согласно варианту задания.
- Динамический массив должен быть сделан в виде шаблона (параметр шаблона – класс для хранения в массиве **template <class T> Array {...}**)
- Сохранять созданные фигуры в динамический массив (переиспользовать от предыдущей лабораторной работы) умных указатели на фигуру (**std::shared_ptr<T[]>**)
- Фигуры должны иметь переопределенные операции копирования, сравнения и приведение к типу double (вычисление площади)
- Вызывать для всего массива общие функции (1-3 см. выше). Т.е. распечатывать для каждой фигуры в массиве геометрический центр, координаты вершин и площадь.
- Необходимо уметь вычислять общую площадь фигур в массиве.
- Удалять из массива фигуру по индексу;
- При увеличении емкости копирования должна использоваться операция перемещения (move) для элементов массива (при перемещении из массива меньшего размера в массив большего размера)
- Программа должна содержать примеры работы с шаблоном массива как для базового типа **Figure*** (например, проверить для **Array<Figure<int>*>**) так и для типов наследников, например Square (**Array<Square<int>>**).

Варианты заданий:

Вариант	Фигура №1	Фигура №2	Фигура №3
1.	Треугольник	Квадрат	Прямоугольник

2.	Квадрат	Прямоугольник	Трапеция
3.	Прямоугольник	Трапеция	Ромб
4.	Трапеция	Ромб	5-угольник
5.	Ромб	5-угольник	6-угольник
6.	5-угольник	6-угольник	8-угольник
7.	6-угольник	8-угольник	Треугольник
8.	8-угольник	Треугольник	Квадрат
9.	Треугольник	Квадрат	Прямоугольник
10.	Квадрат	Прямоугольник	Трапеция
11.	Прямоугольник	Трапеция	Ромб
12.	Трапеция	Ромб	5-угольник
13.	Ромб	5-угольник	6-угольник
14.	5-угольник	6-угольник	8-угольник
15.	6-угольник	8-угольник	Треугольник
16.	8-угольник	Треугольник	Квадрат
17.	Треугольник	Квадрат	Прямоугольник
18.	Квадрат	Прямоугольник	Трапеция
19.	Прямоугольник	Трапеция	Ромб
20.	Трапеция	Ромб	5-угольник
21.	Ромб	5-угольник	6-угольник
22.	5-угольник	6-угольник	8-угольник
23.	6-угольник	8-угольник	Треугольник
24.	8-угольник	Треугольник	Квадрат
25.	Треугольник	Квадрат	Прямоугольник
26.	Квадрат	Прямоугольник	Трапеция
27.	Прямоугольник	Трапеция	Ромб
28.	Трапеция	Ромб	5-угольник
29.	Ромб	5-угольник	6-угольник
30.	5-угольник	6-угольник	8-угольник
31.	6-угольник	8-угольник	Треугольник
32.	8-угольник	Треугольник	Квадрат
33.	Треугольник	Квадрат	Прямоугольник
34.	Квадрат	Прямоугольник	Трапеция
35.	Прямоугольник	Трапеция	Ромб
36.	Трапеция	Ромб	5-угольник