

Лабораторная работа № 03

Тема: Наследование, полиморфизм

Цель:

- Изучение механизмов работы с наследованием в C++;
- Изучение механизма перегрузки операций

Порядок выполнения работы

- Ознакомиться с теоретическим материалом.
- Получить у преподавателя вариант задания.
- Реализовать задание своего варианта в соответствии с поставленными требованиями.
- Написать Unit-тесты с использованием Google Test.
- Создать репозиторий на GitHub.
- Отправить файлы лабораторной работы в репозиторий.
- 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 программу на языке C++ согласно варианту задания. Программа на C++ должна собираться с помощью системы сборки CMake. Программа должна получать данные из стандартного ввода (`std::cin`) и выводить данные в стандартный вывод (`std::cout`).

Необходимо зарегистрироваться на GitHub и создать репозиторий для задания лабораторных работ.

Преподавателю необходимо предъявить ссылку на публичный репозиторий на Github. Необходимо реализовать функцию согласно варианту задания. Функция должна быть помещена в отдельный файл.

Разработать классы согласно варианту задания, классы должны наследоваться от базового класса `Figure`. Фигуры являются фигурами вращения.

Все классы должны поддерживать набор общих методов:

1. Вычисление геометрического центра фигуры вращения;
2. Вывод в стандартный поток вывода `std::cout` координат вершин фигуры через перегрузку оператора `<<` для `std::ostream`;
3. Чтение из стандартного потока данных фигур через перегрузку оператора `>>` для `std::istream`
4. Вычисление площади фигуры через перегрузку оператора приведения к типу `double`;

Создать программу, которая позволяет:

- Вводить из стандартного ввода `std::cin` фигуры, согласно варианту задания.
- Сохранять созданные фигуры в динамический массив (по аналогии с предыдущей лабораторной работой `Array`) указатели на фигуру (`Figure*`)
- Фигуры должны иметь переопределенные операции копирования (`=`), перемещения (`=`) и сравнения (`==`)

- Вызывать для всего массива общие функции (1-3 см. выше). Т.е. распечатывать для каждой фигуры в массиве геометрический центр и площадь.
- Необходимо уметь вычислять общую площадь фигур в массиве.
- Удалять из массива фигуру по индексу;

Варианты заданий (выпуклые равносторонние фигуры вращения):

Вариант	Фигура №1	Фигура №2	Фигура №3
1.	Треугольник	Квадрат	Прямоугольник
2.	Квадрат	Прямоугольник	Трапедия
3.	Прямоугольник	Трапедия	Ромб
4.	Трапедия	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	Трапедия
11.	Прямоугольник	Трапедия	Ромб
12.	Трапедия	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	Трапедия
19.	Прямоугольник	Трапедия	Ромб
20.	Трапедия	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	Трапедия
27.	Прямоугольник	Трапедия	Ромб
28.	Трапедия	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

Отчет

1. Код программы на языке C++.
2. Ссылка на репозиторий на GitHub.
3. Набор testcases на Google Test.