МИНИСТЕРСТВО НАУКИ и высшего образования

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное учреждение

высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

(ФГАОУ ВО «СПбПУ»)

**Институт среднего профессионального образования**

**Отчёт по лабораторной работе № 4**

**по учебной дисциплине «Системное программирование»**

**Тема: «Перегрузка операций»**

Выполнил(а) студент(ка)

специальности 09.02.07

Информационные системы и

программирование

IV курса группы 42919/7

Побирчев Никита Андреевич

Преподаватель

Молькова Лолита Юрьевна

Санкт-Петербург,

2024

**Лабораторная работа №4**

**Цель работы:**

Изучить механизм перегрузки операций.

**Задание:**

Все классы следует наделить конструкторами, деструктором. Необходимо явно реализовать конструктор копирования и перегрузить оператор присваивания. Необходимо подготовить демонстрацию по работе перегруженных для класса операторов.

**Индивидуальное задание:**

Создать класс КОЛЬЦО (геометрическая фигура на плоскости) в прямоугольной системе координат OXY с центром в точке O(0,0), которая определяется двумя радиусами (R1, R2, R2>R1>0). Реализовать метод вывода на экран информации о кольце. Перегрузить бинарные операторы: пересечение двух колец (\*), объединение двух колец (+), несимметрическая разность двух колец (-). Следует учесть, что результатом выполнения оператора может быть получено кольцо с равными радиусами (R1=R2), в этом случае следует сообщить об ошибке.

**Ход работы:**

Создал класс Ring, инициализировал его свойства (Рисунок 1).

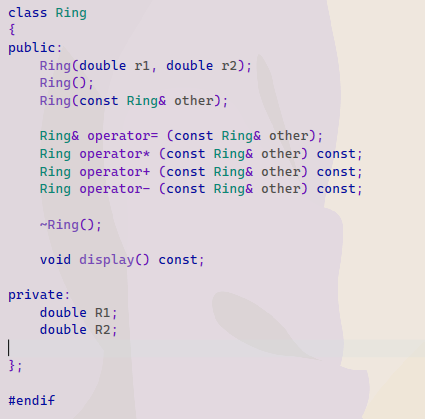


Рисунок 1 - Инициализация класса MyString

Добавил конструктор по умолчанию, а также конструктор запрашивающий все аргументы (Рисунок 2).

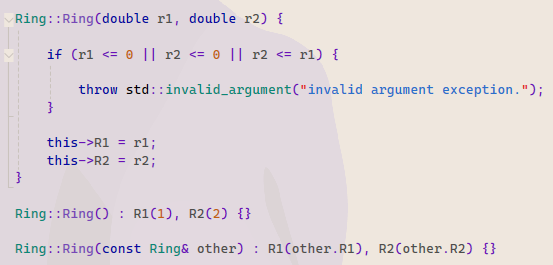


Рисунок 2 – Конструкторы

Добавил оператор присваивания и умножения (Рисунок 3).

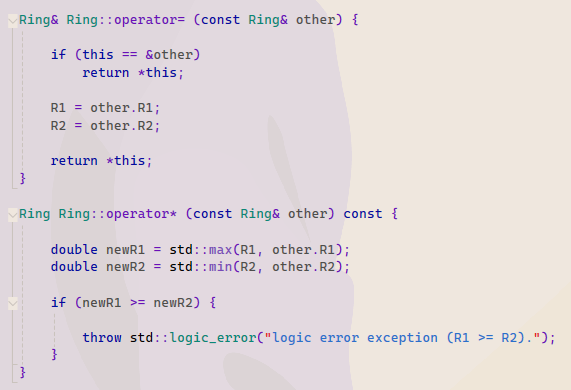


Рисунок 3 – Оператор = и \*

Добавил оператор вычитания и сложения (Рисунок 4).

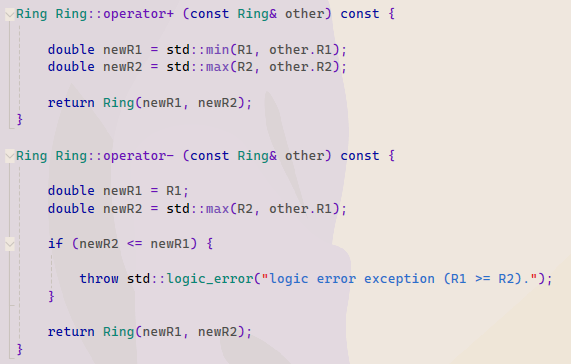


Рисунок 4 – Оператор + и -

Добавил метод для вывода информации о состоянии объекта в консоль (Рисунок 5).

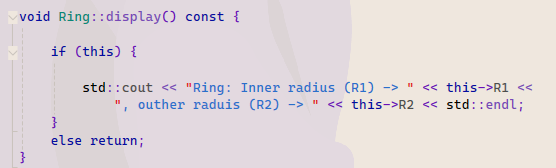


Рисунок 5 - Метод для вывода информации о состоянии объекта

Деструктор по умолчанию (Рисунок 7).



Рисунок 7 - Деструктор

Далее в main.cpp, пользователю необходимо ввести внутренний и внешний диаметр колец. Далее на практике используются перегруженные операторы (Рисунок 8).

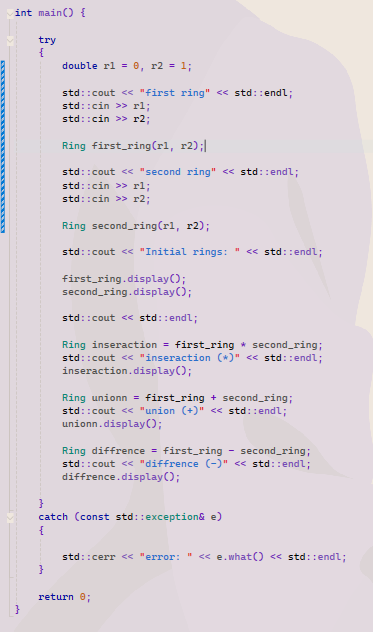


Рисунок 8 - Функция main.

Вывод результата работы программы в консоль (Рисунок 9).\

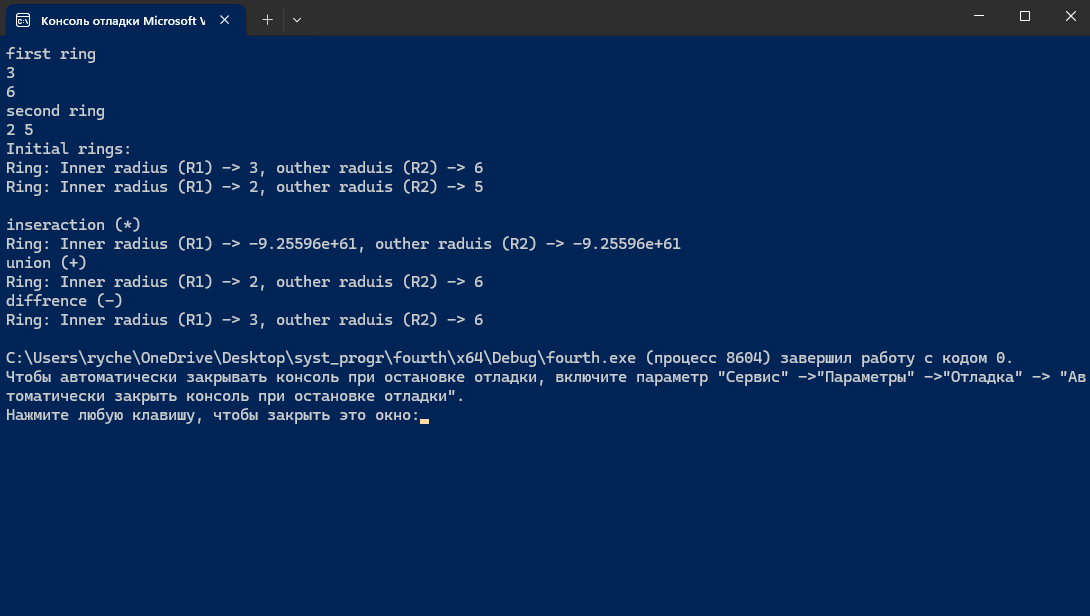


Рисунок 9 - Вывод в консоль

**Контрольные вопросы**

1. **Для чего в C++ применяется перегрузка операций**

Перегрузка операций используется для того, чтобы определить собственное поведение операторов.

1. **Истинно ли следующее утверждение: операция >= может быть перегружена?**

Да, это утверждение истинно.

1. **Сколько аргументов требуется для определения перегруженной унарной операции?**

Для унарной операции требуется 1 аргумент.

1. **Сколько аргументов требуется для определения перегруженной бинарной операции?**

Для бинарной операции требуется 2 аргумента.

**5. Чем отличается действие операции ++ в префиксной форме от её действия в постфиксной форме?**

Префиксная форма сразу увеличивает и возвращает значение. Префиксная форма увеличивает значение, возвращая значение до увеличения.

1. **Истинно ли следующее утверждение: перегруженная операция всегда требует на один аргумент меньше, чем количество операндов?**

Методы класса получают неявный аргумент this, поэтому для унарной операции нужен 1 явный аргумент, а для бинарной — 2.

**7. Когда перегружается операция арифметического присваивания, то результат**

b. Передается объекту слева от операции

**8. Истинно ли следующее утверждение: компилятор не выдаст сообщение об ошибке, если вы перегрузите операцию \* для выполнения деления?**

Да, данное утверждение верно.

**9. Существуют ли операции, которые нельзя перегружать?**

* Тернарный оператор ?:.
* Оператор разрешения области видимости ::.
* Оператор выбора члена через указатель на объект .\*.
* Оператор выбора члена через указатель на указатель ->\*.
* Оператор sizeof.
* Оператор typeid.
* Оператор приведения типа (static\_cast, dynamic\_cast, const\_cast, reinterpret\_cast).