Пермский филиал федерального государственного автономного  
образовательного учреждения высшего образования  
«Национальный исследовательский университет  
«Высшая школа экономики»

*Факультет социально-экономических и компьютерных наук*

Мальшаков Александр Аркадьевич

**ЛАБОРАТОРНАЯ РАБОТА №9**

*Классы и объекты*

студента образовательной программы «Разработка информационных систем для бизнеса» по направлению подготовки 38.03.05-Бизнес - информатика

Руководитель

Старший преподаватель кафедры ИТБ \_\_\_\_\_\_\_\_\_\_\_\_

Волков С. А.

Пермь, 2024 год

**Оглавление**

[Постановка задачи 3](#_Toc190032779)

[Общая 3](#_Toc190032780)

[Часть 1. 3](#_Toc190032781)

[Часть 2. 3](#_Toc190032782)

[Часть 3 3](#_Toc190032783)

[Вариант №9 4](#_Toc190032784)

[Диаграмма классов 5](#_Toc190032785)

[Код программы и тестов 6](#_Toc190032786)

[Анализ покрытия кода тестами 7](#_Toc190032787)

# Постановка задачи

## Общая

### Часть 1.

1. Реализовать определение нового класса/типа данных. Необходимо реализовать не менее 2 конструкторов: без параметров, с параметрами.
2. Для демонстрации работы с объектами написать главную функцию, в которой создаются объекты класса и выводится информация, которая содержится в атрибутах.
3. Написать функцию, выполняющую указанное в варианте действие. Рассмотреть два варианта реализации функции:
4. статическую функцию;
5. метод класса.
6. В основной программе продемонстрировать работу функций, объяснить разницу между статической и нестатической функциями.
7. Используя статическую компоненту класса подсчитать количество созданных в программе объектов.

### Часть 2.

1. Добавить к реализованному классу указанные в варианте перегруженные операции.
2. Добавить метод public override bool Equals(object obj) {} для сравнения двух объектов реализованного класса.
3. Написать демонстрационную программу, в которой создаются объекты пользовательских классов и выполняются указанные операции.

### Часть 3

1. Реализовать класс-коллекцию, полем которого является одномерный массив из элементов заданного в варианте типа.
2. В классе реализовать:

* конструктор без параметров;
* конструктор с параметрами, заполняющий элементы случайными значениями;
* конструктор копирования, позволяющий создать копию коллекции, которая передается в конструктор как параметр, д.б. реализовано глубокое копирование.
* метод для просмотра элементов массива.

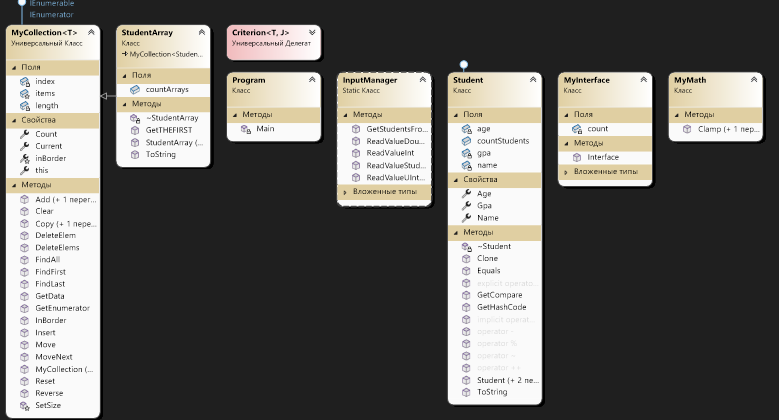
1. Реализовать индексатор для доступа к элементам коллекции. Предусмотреть проверку при выходе индекса за пределы массива и обработку этой ситуации с помощью исключительных ситуаций.
2. Написать демонстрационную программу, позволяющую создать массив разными способами (ручной ввод и случайная генерация) и распечатать элементы массива. Создать новую коллекцию на основе существующей, показать, что выполнено глубокое копирование.
3. В демонстрационной программе показать 4 варианта работы индексатора.
4. Написать функцию в классе Program для выполнения указанного в варианте задания, т.е. функция должна перебирать коллекцию и обрабатывать ее элементы, используя перегруженные операции пользовательского класса.
5. Подсчитать количество созданных объектов и созданных коллекций.

## Вариант №9

1. Реализовать класс Student с 2-я конструкторами, 3 полями – возраст, средний балл и имя.
2. Добавить в этот класс функция которая возвращает строку сравнения 2-х студентов.
3. Определить унарные операции ~ и ++.
4. Определить операции явного приведения к типу int и неявного приведения к типу bool.
5. Определить бинарные операции Student % string, Student – double.
6. Создать класс коллекцию хранящий массив студентов, определить доступ к студентам по индексу, а также функцию возвращающую самого старшего студента с средний балл > 8.
7. Создать демонстрационную программу для всех вышеуказанных функций.

# Диаграмма классов

На рисунке ниже представлена диаграмма классов (см. рисунок 1).



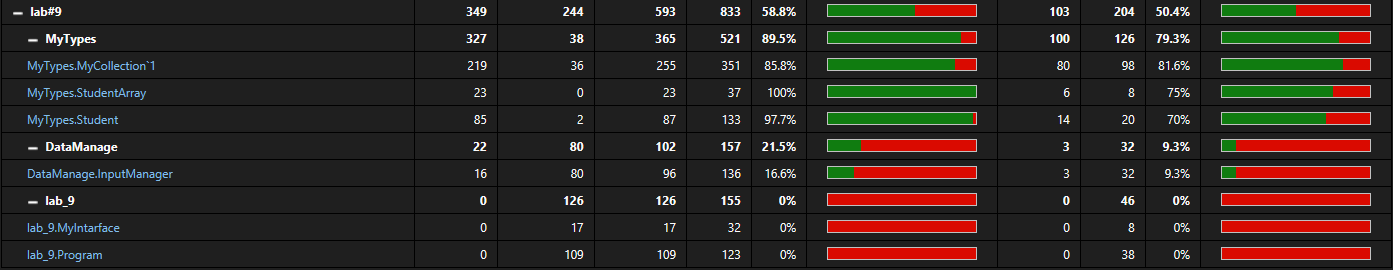
***Рисунок 1 – диаграмма классов***

# Код программы и тестов

Ссылка на GitHub с кодом программы и тестов <https://github.com/Poruch/LabWorks.git>.

# Анализ покрытия кода тестами

Ниже на рисунке видно покрытие тестами кода (см. рисунок 2). Классы с бизнесс логикой находяться в пространстве имен MyTypes и покрыты на 79% остальные классы не тестировались так как это в них идёт работа с выводом/вводом.



***Рисунок 2 – покрытие кода тестами***