|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет социально-экономических и компьютерных наук* |
|  |
| Ханжин Александр Евгеньевич  **ЛАБОРАТОРНАЯ РАБОТА №9**  студента образовательной программы «Разработка информационных систем для бизнеса» по направлению подготовки *09.03.04 Программная инженерия*   |  |  | | --- | --- | |  | Преподаватель кафедры ИТБ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Марквирер В.Д. |   Пермь, 2024 |

**Оглавление**

[Задача 1 3](#_Toc157889102)

[1.1 Постановка задачи 3](#_Toc157889103)

[1.2 Диаграмма классов 5](#_Toc157889104)

[1.3 Программа (листинг) 6](#_Toc157889105)

[1.4 Тестирование программы 7](#_Toc157889106)

# Задача 1

## Постановка задачи

В данной лабораторной работе рассматривается тема классов в языке программирования C#. Классы являются основным строительным блоком объектно-ориентированного программирования и позволяют создавать пользовательские типы данных. В ходе выполнения работы мы изучим, как создавать классы, определять их поля, свойства и методы. Применение классов позволяет создавать более гибкие и модульные программы, повышая читаемость и удобство использования кода. Данная лабораторная работа позволит нам познакомиться с основными концепциями и возможностями классов в языке C# и применить полученные знания на практике.

В 9 варианте предлагается реализовать класс, реализующий создание студента с учетом его имени, возраста и средней оценки GPA. Программа должная решать множество проблем и выполнять следующие функции:

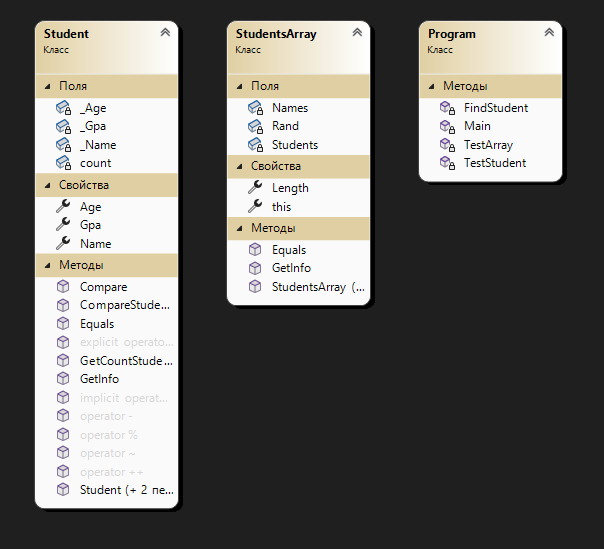
1. В отдельном файле требуется реализовать определение нового класса/типа данных. В этой реализации должны присутствовать закрытые атрибуты, свойства, конструкторы и методы инициализации и вывода атрибутов. Обязательно необходимо создать три конструктора: без параметров, с параметрами и конструктор копирования.
2. Для демонстрации работы с объектами нужно написать основную функцию, в которой будут создаваться объекты класса и выводиться информация, содержащаяся в их атрибутах.
3. Требуется написать функцию, выполняющую заданное действие. Эту функцию можно реализовать двумя способами: в виде статической функции и метода класса. В основной программе нужно продемонстрировать работу этих функций и объяснить разницу между статическими и нестатическими функциями.
4. С использованием статической компоненты класса необходимо подсчитать количество созданных в программе объектов.
5. Выполнить перегрузку операций для реализованного класса в соответствии с вариантом задания.
6. Унарные операции: привести имя студента к формату, в котором первая буква заглавная, а остальные строчные и увеличить возраст студента на 1.
7. Операции приведения типа: явное приведение к типу int, результатом является номер курса, на котором обучается студент. Если студенту 18 лет, то он обучается на первом курсе. Если студенту больше 22 лет, возвращается -1 как флаг невозможности точного определения номера курса неявное приведение к типу bool, результатом является true, если gpa < 6 (студент скорее всего имеет удовлетворительные оценки), иначе - false.
8. Бинарные операции: % Student s, string newName – результатом является новый студент с тем же возрастом и gpa, но другим именем и Student s, double d – (правосторонняя операция) результатом является тот же студент, но с уменьшенным gpa на заданное число d. При этом необходимо учесть, что gpa не может быть меньше 0.
9. Реализовать метод public override bool Equals(object obj) {} для сравнения двух объектов данного пользовательского класса. Без этого метода unit-тесты не будут работать.
10. Написать программу-демонстрацию, в которой создаются объекты указанного пользовательского класса и выполняются требуемые операции.

Помимо этого, нужно реализовать класс-коллекцию, объединяющую объекты класса студента в 1 массив. В классе нужно реализовать:

1. Конструктор без параметров;
2. Конструктор с параметрами, заполняющий элементы случайными значениями;
3. Конструктор с параметрами, позволяющий заполнить массив элементами, заданными пользователем с клавиатуры;
4. Конструктор копирования, позволяющий создать копию коллекции, которая передается в конструктор как параметр, д.б. реализовано глубокое копирование.
5. Метод для просмотра элементов массива.

## Диаграмма классов

Диаграмма классов является одним из основных инструментов объектно-ориентированного моделирования и позволяет описать архитектуру программного обеспечения. Она помогает разработчикам понять, какие классы и объекты присутствуют в системе, а также как они взаимодействуют между собой. Диаграмма классов лабораторной работы представлена на рисунке 1.2.1



***Рис. 1.2.1. Диаграмма классов.***

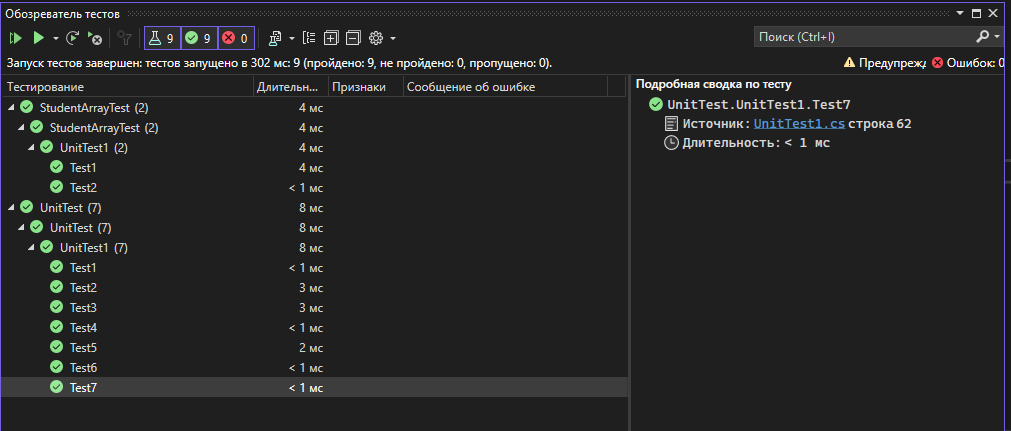
## Программа (листинг)

Полный листинг программы можно найти на вебсайте github по ссылке: <https://github.com/Asklit/C-lab9/tree/main/lab9>. Тесты класса Student можно найти по ссылке <https://github.com/Asklit/C-lab9/tree/main/StudentTest> и класса StudentsArray https://github.com/Asklit/C-lab9/tree/main/StudentsArrayTest

## Тестирование программы

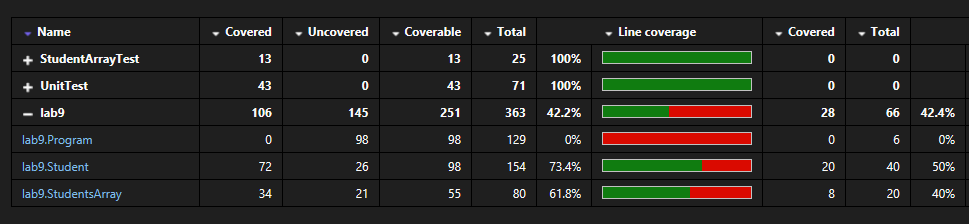
При тестировании кода имеет большое значение обеспечение его правильной работы и предотвращение возможных ошибок. Тестирование играет важную роль в разработке программного обеспечения, позволяя выявить проблемные места и убедиться в корректности функционирования кода. В этом тестировании кода мы будем искать потенциальные баги, проверять работу основных функций и убеждаться в соответствии ожидаемого результата с фактическим. Тестирование кода позволит не только обнаружить возможные ошибки, но и повысить общую качество программного продукта.

Обзор написанных тестов представлен на рисунке 1.4.1.



***Рис. 1.4.1. Обзор написанных тестов для программы.***

В ходе написания тестов было достингнуто покрытие текстами кода 60-70%. Таким образом была тестами была покрыта вся «бизнес логика» программы.



***Рис. 1.5.2. Покрытие кода тестами.***