|  |  |
| --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  РОССИЙСКОЙ ФЕДЕРАЦИИ  Федеральное государственное автономное  образовательное учреждение высшего образования  «Пермский государственный национальный исследовательский университет» | |
|  | Институт компьютерных наук и технологий |
| ОТЧЕТ  по лабораторной работе **№7**  по дисциплине Языки программирования  «**ООП**» | |
|  | Работу выполнила студентка  Группы ПМИ 1-2, курс 2  Института компьютерных наук и технологий  **Арланова Анна Андреевна** |
|  | Работу принял:  ассистент кафедры Информационных технологий  Ракина В.Д. |
| Пермь, 2023 | |

Оглавление

[1 Постановка задачи 3](#_Toc152648357)

[2 Проектирование 4](#_Toc152648358)

[2.1 Класс со строковым полем 4](#_Toc152648359)

[2.2 Класс Money 6](#_Toc152648360)

[3 Тестирование 8](#_Toc152648361)

[3.1 Класс со строковым полем 8](#_Toc152648362)

[3.2 Класс Money 10](#_Toc152648363)

# Постановка задачи

1. Решить задачу, используя класс List. Составить программу, которая в списке L заменяет первое вхождение списка L1 (если такое есть) на список L2
2. Решить задачу, используя класс LinkedList. Сортировка элементов списка по возрастанию.
3. Решить задачу, используя класс HashSet. Есть перечень компьютерных игр. Студенты группы играют в какие-либо из этих игр. Известно для каждого студента, в какие игры он играет. Определить:

• в какие игры из перечня играют все студенты группы;

• в какие игры из перечня играют некоторые студенты группы;

• в какие игры из перечня не играет ни один из студентов группы?

1. Решить задачу, используя класс HashSet. Файл содержит текст на русском языке. Напечатать в алфавитном порядке все глухие согласные буквы, которые не входят хотя бы в одно слово.
2. Решить задачу, используя класс Dictionary (или класс SortedList). Решить текстовую задачу с использованием словаря (входные данные читать из текстового файла). На городской олимпиаде по информатике участникам было предложено

выполнить 3 задания, каждое из которых оценивалось по 25-балльной шкале. Известно, что общее количество участников первого тура олимпиады не превосходит 250 человек. На вход программы подаются сведения о результатах олимпиады. В первой строке вводится количество участников N. Далее следуют N строк, имеющих следующий формат:

<Фамилия> <Имя> <Баллы>

Здесь <Фамилия> – строка, состоящая не более чем из 20 символов; <Имя> –строка, состоящая не более чем из 15 символов; <Баллы> – строка, содержащая три целых числа, разделенных пробелом, соответствующих баллам, полученным участником за каждое задание первого тура. При этом <Фамилия> и <Имя>, <Имя> и <Баллы> разделены одним пробелом. Примеры входных строк:

Петрова Ольга 25 18 16

Калиниченко Иван 14 19 15

Напишите программу, которая будет выводить на экран фамилию и имя участника, набравшего максимальное количество баллов. Если среди остальных участников есть ученики, набравшие такое же количество баллов, то их фамилии и имена также следует вывести. При этом имена и фамилии можно выводить в произвольном порядке.

# Проектирование

Разделила программу на 3 файла:

1. FileFunction.cs - все необходимые функции для решения задач
2. UserInput.cs – проверка корректности пользовательского ввода
3. Program.cs – вывод менюшки и выбор задания

## Замена первого вхождения списка

Суть проста, создаем три списка L, L1 и L2, при помощи метода из UserInput.cs, который позволяет формировать строковые списки, где каждая строчка с консоли это элемент для списка. Пустая строка означает конец списка. Далее выводим получившиеся списки при помощи универсального метода вывода списков. Затем вызываем метод для замены первого вхождения списка на другой список. Метод ищет первое вхождение списка L1 в списке L. Если такое вхождение найдено, он удаляет элементы, соответствующие L1 из исходного списка L и заменяет их на элементы списка L2 в том же месте, где были найдены элементы, соответствующие L1. В случае, если L1 не найдено в L, метод ничего не изменяет и просто возвращает исходный список L. Затем выводим конечный результат, список L.

## Сортировка элементов списка по возрастанию

Создаем связанный список почти аналогично списку из первого заданию. Выводится получившийся результат. Сортировка реализована через универсальный метод, где наш связанный список преобразуется в обычный список и через стандартный метод списка сортируется и возвращается обратно связанным списком. Далее выводится получившийся результат.

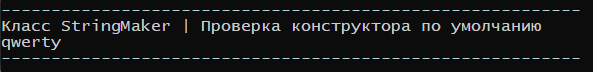
## Перечень компьютерных игр

No comment

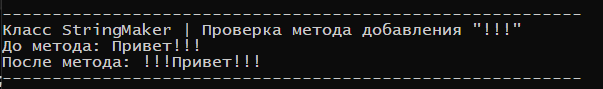
## Глухие согласные буквы, которые не входят в слово

# Тестирование

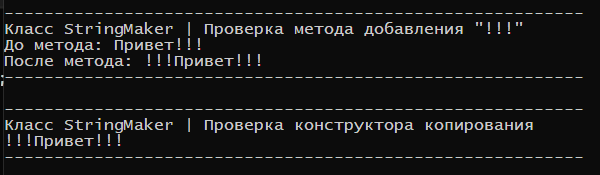
## Класс со строковым полем



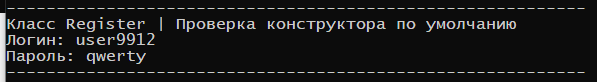
1ый тест



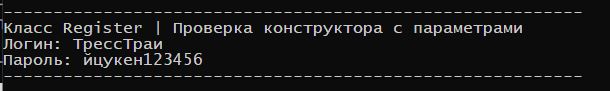
2ой тест



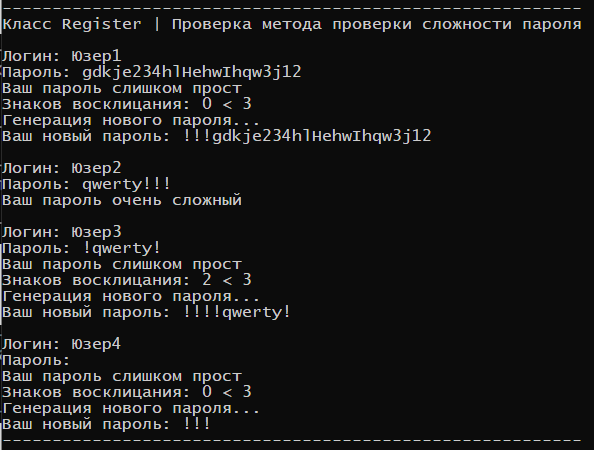
3ий тест



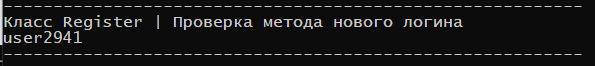
4ый тест



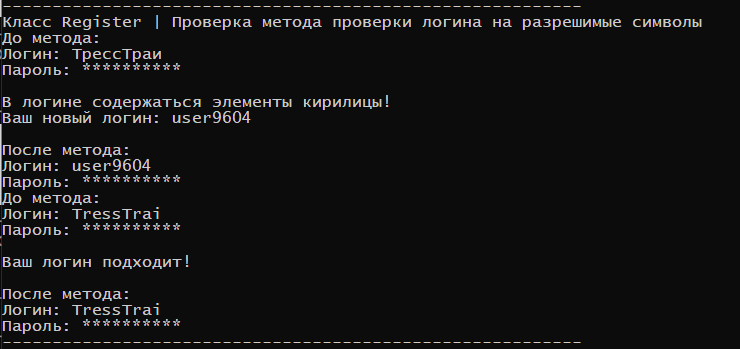
5ый тест



6ой тест



7ой тест

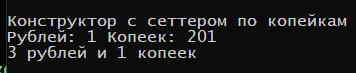


8ой тест

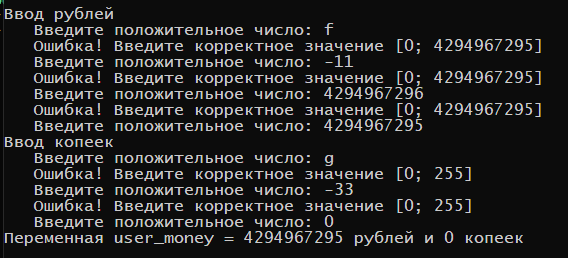
## Класс Money



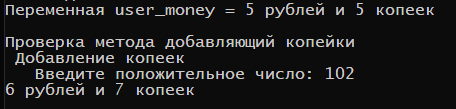
1ый тест



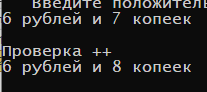
2ой тест



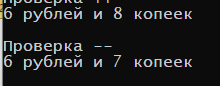
3ий тест



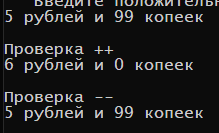
4ый тест



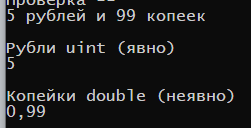
5ый тест



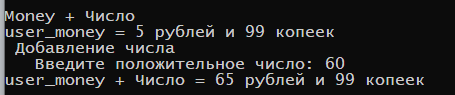
6ой тест



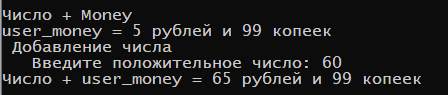
7ой тест



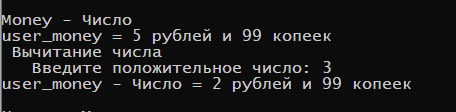
8ой тест



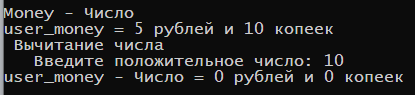
9ый тест



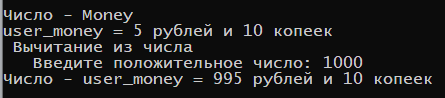
10ый тест



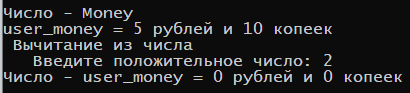
11ый тест



12ый тест



13ый тест



14ый тест