МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное бюджетное образовательное учреждение высшего образования

**«МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)»**

Кафедра «Высшая математика»

**ПРАКТИЧЕСКАЯ РАБОТА № 6**

по дисциплине

«Структуры и алгоритмы обработки данных»

на тему:

«Работа со списками»

|  |  |
| --- | --- |
|  | **Выполнил:**  Учебная группа: 1бПМ  ФИО: Жиленко А.А.  Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **Руководитель практической работы:**  Должность: старший преподаватель  Звание: б/з  ФИО: Кутейников И.А.  Подпись \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 \_\_\_\_ г. |

Москва 2023 г.

Содержание

1. Цель и постановка задачи
2. Алгоритм решения
3. Результат решения
4. Заключение

**Цель и постановка задачи**

**Цель работы:** изучение работы со списками и классами ООП.

**Задачи:**

1. Реализовать в Java класс компьютер. Предусмотреть не менее 3 параметров, одним из которых является объект другого класса, 2 методов и 2 конструкторов (включая конструктор по умолчанию). Предусмотреть счетчик экземпляров классов.

Предусмотреть для классов, являющихся полями не менее 2 параметров, 2 методов и 2 конструкторов (включая конструктор по умолчанию).

1. Реализовать хранение объектов классов в виде списка.

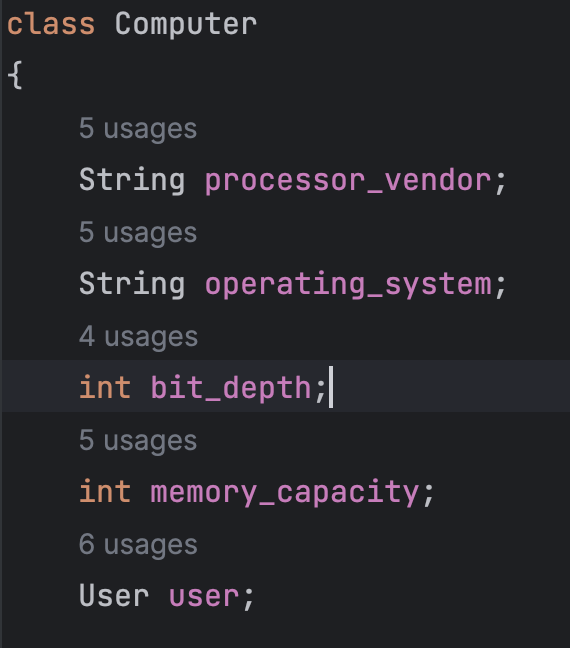
Предусмотреть следующие операции над списком:

* 1. Добавления элемента (доп. нескольких элементов) в конец или в определенное место списка
  2. Удаление элемента (доп. нескольких элементов) в конце или в определенном месте списка
  3. Изменение определенного элемента списка
  4. Поиск и вывод номера и информации введенного элемента по разным полям
  5. Вывод на экран всех объектов списка или определенного элемента списка
  6. Вывод числа элементов в списке.

**Алгоритм решения**

Задача 1:

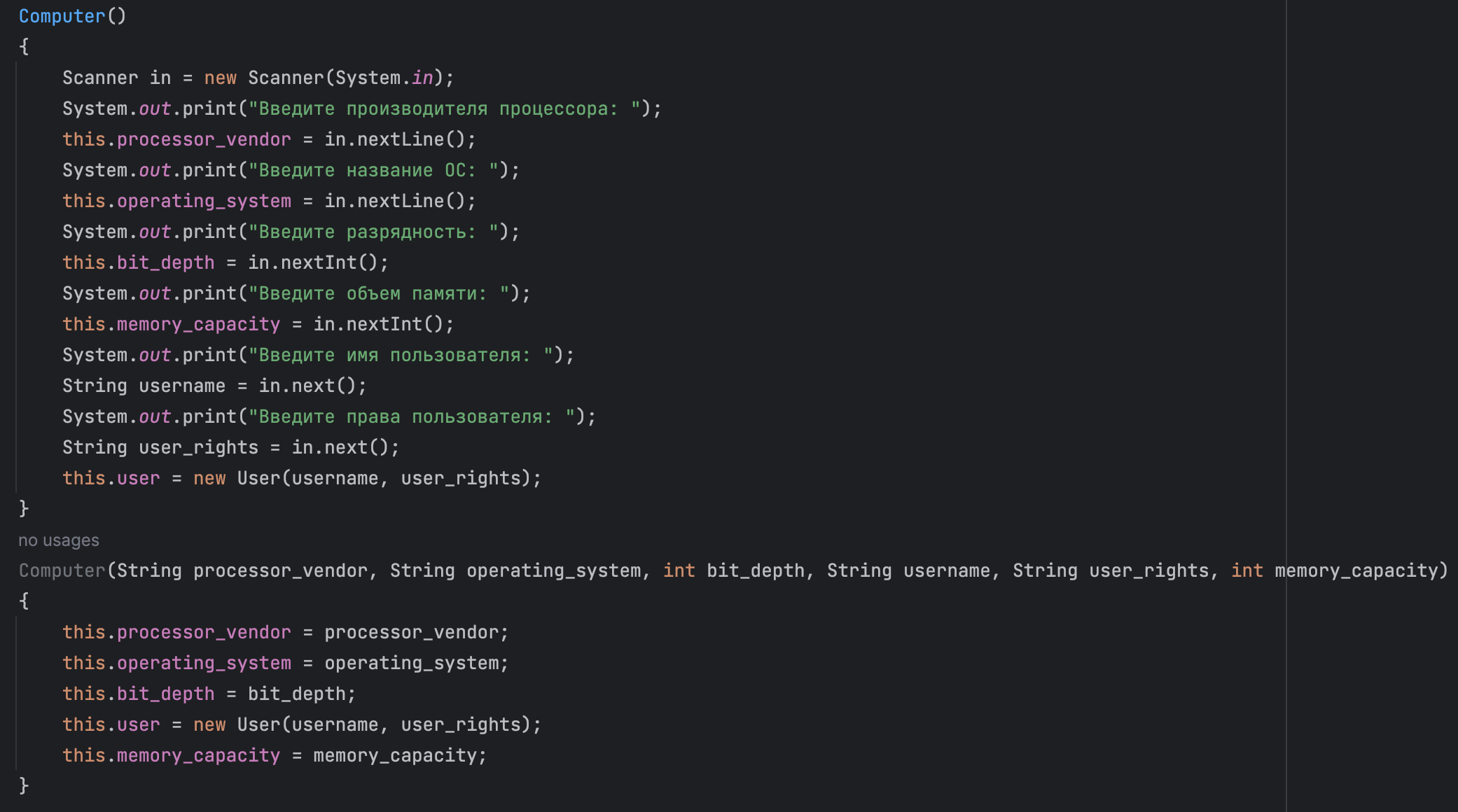
Класс компьютер с полями: processor\_vendor – производитель процессора, operating\_system – операционная система, bit\_depth - разрядность, memory\_capacity – объем памяти, user – пользователь, объект класса User.



Методы класса:



Конструкторы:



Класс User:

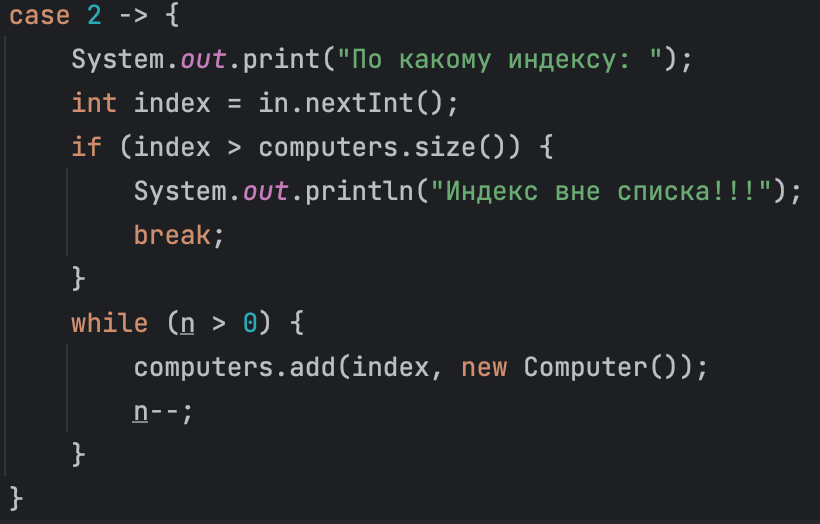


Задача 2.a:

У пользователя спрашивается кол-во объектов которое будет добавлено и с какого индекса будут добавляться, после чего добавление объекта происходит в список ArrayList<Computer> computers методом add(new Computer())

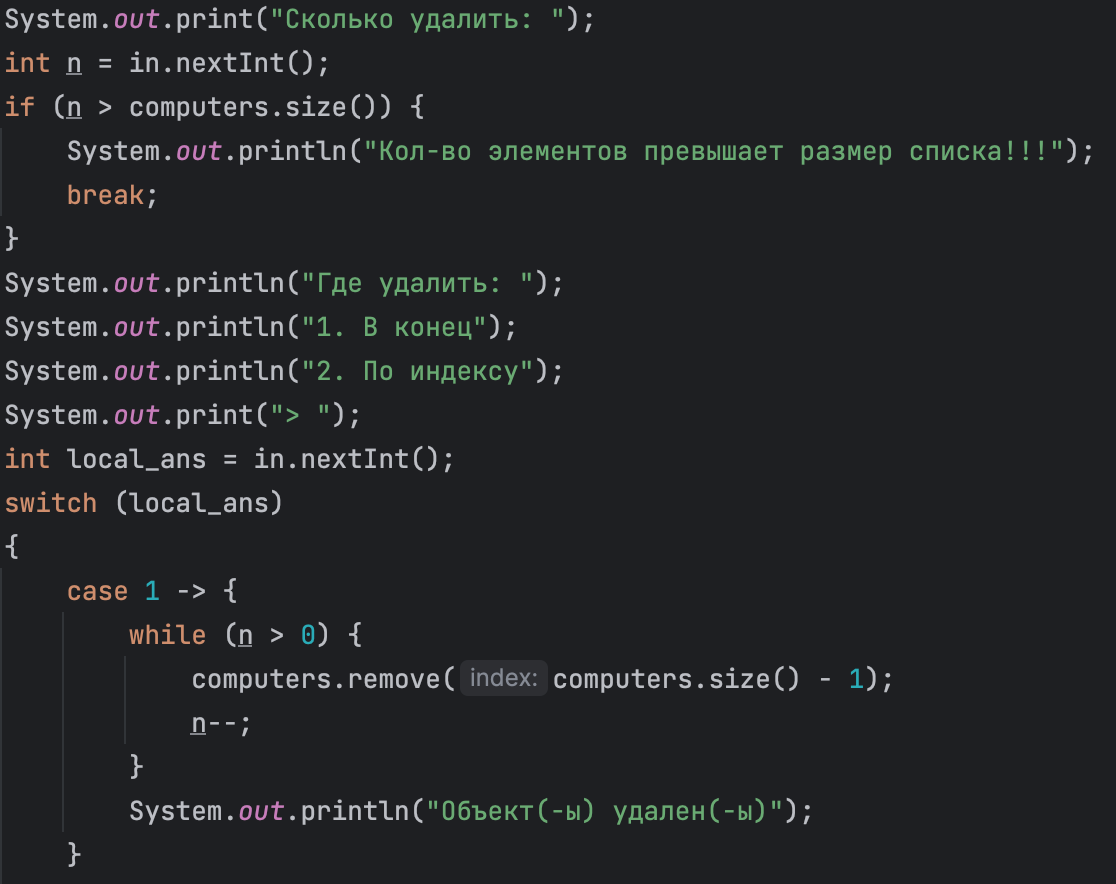


Если пользователь ввёл несуществующий индекс, то соответствующее сообщение выводится на экран

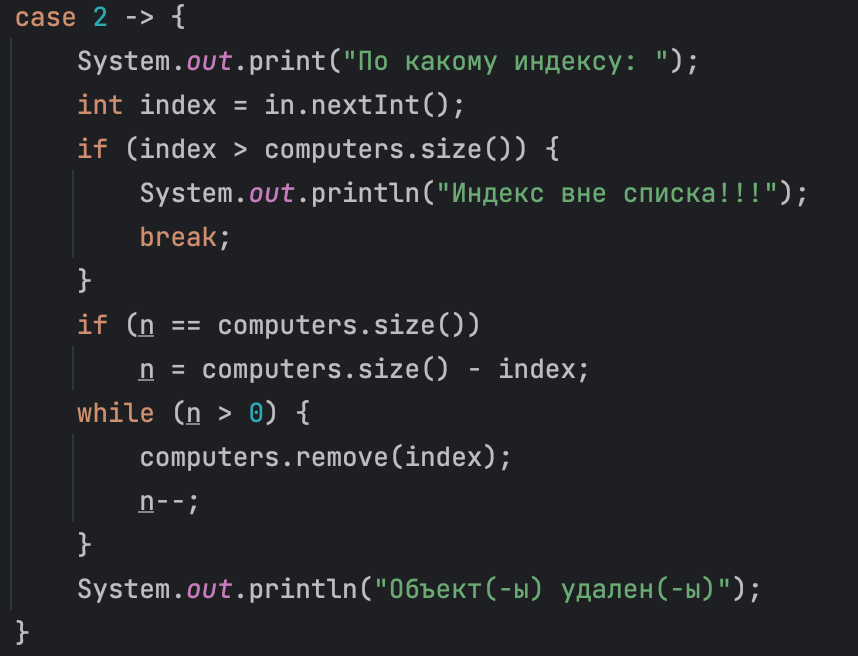


Задача 2.b:

Сначала у пользователя спрашивается сколько объектов нужно удалить, также проходит проверка, если кол-во удаляемых объектов превышает кол-во объектов в списке, то на экран пользователя выводится соответствующее сообщение.

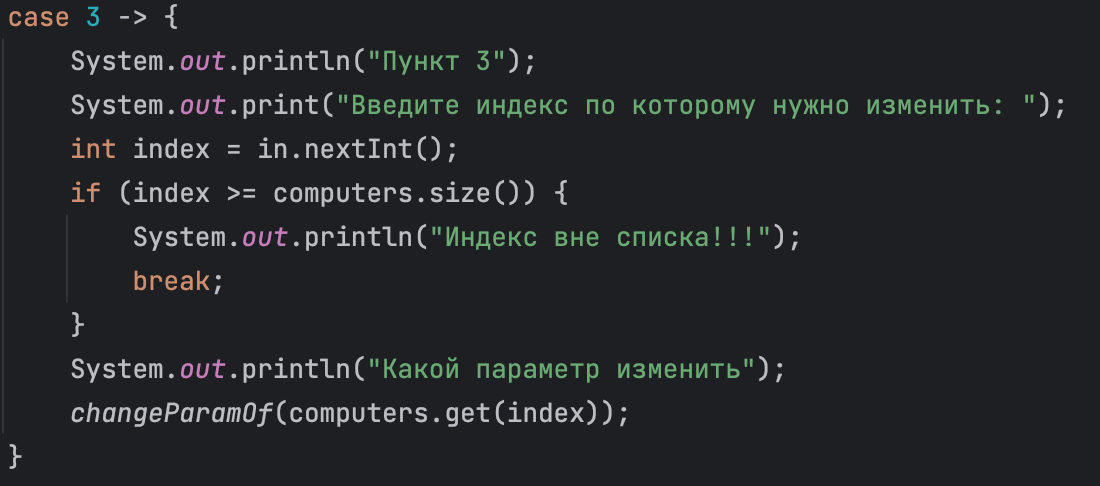


Также удаление может происходить с какого-то индекса, если n равен длине списка, то удаляем объекты до конца списка вне зависимости от введенного индекса.



Задача 2.c:

Спрашиваем у пользователя индекс объекта который нужно изменить, проверяем индекс на существование в списке, вызываем метод changeParamOf(), который спрашивает у пользователя какое поле изменить и заменяет при помощи сеттеров класса.

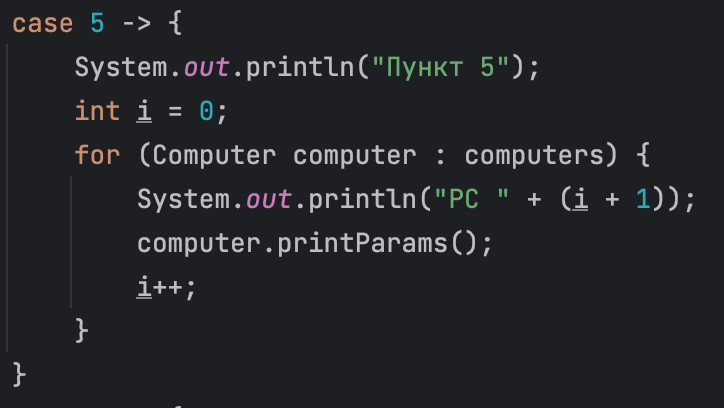


Задача 2.d:

Поиск по параметрам объекта выполняется методом findComputerBy(). У пользователя спрашивается по какому полю искать и какой ключ поиска. Поиск выполняется последовательным методом.

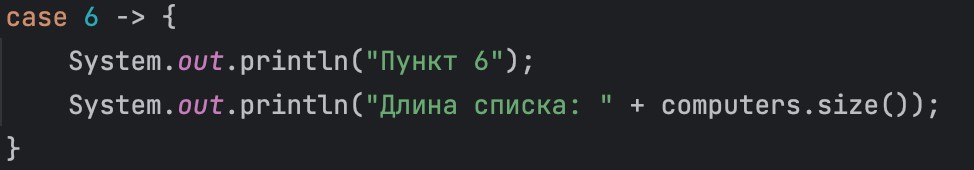
Задача 2.e:

Вывод происходит циклом foreach и методом printParams()



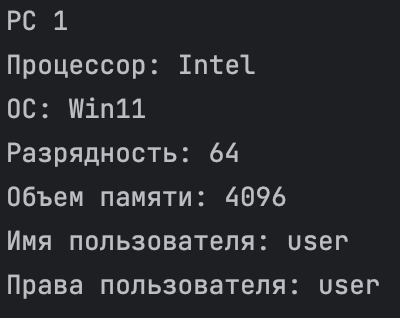
Задача 2.f:

Длину списка получаем методом size()

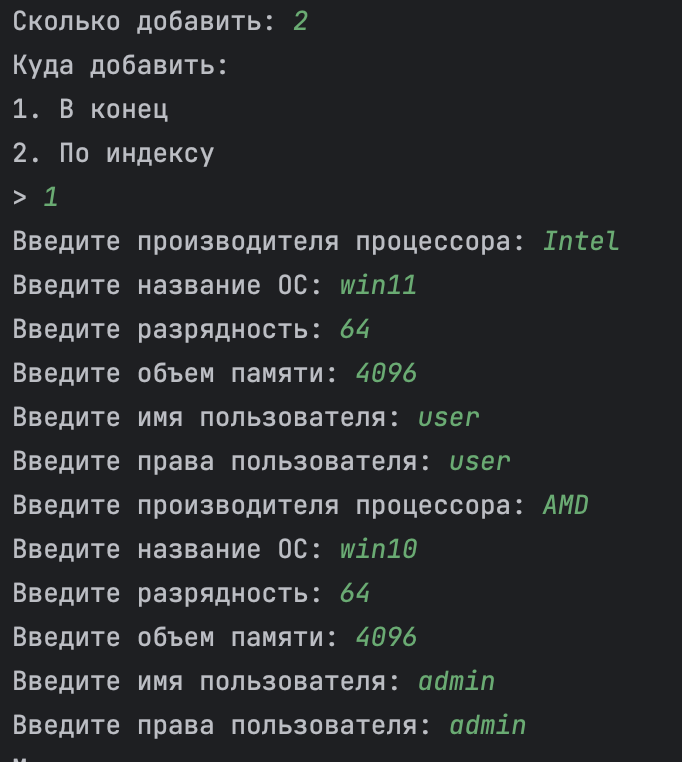


**Результат решения**

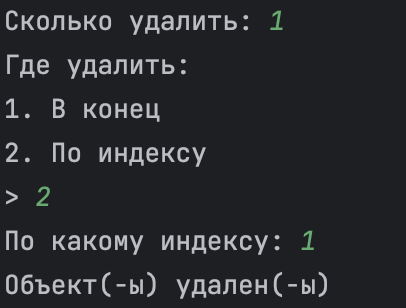
Задача 1:



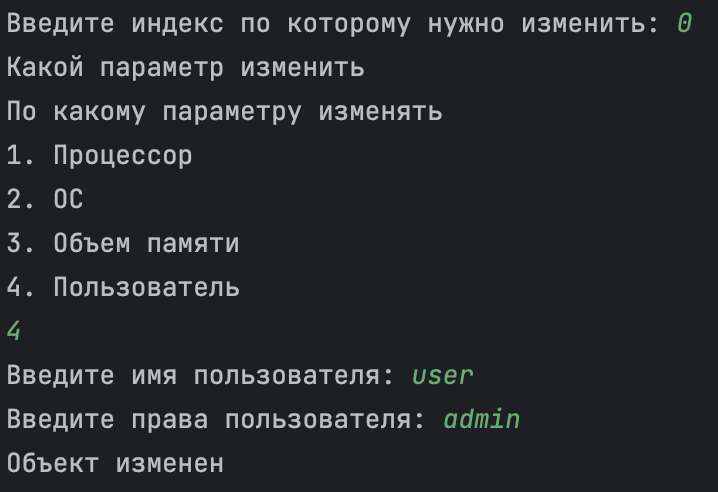
Задача 2.a:



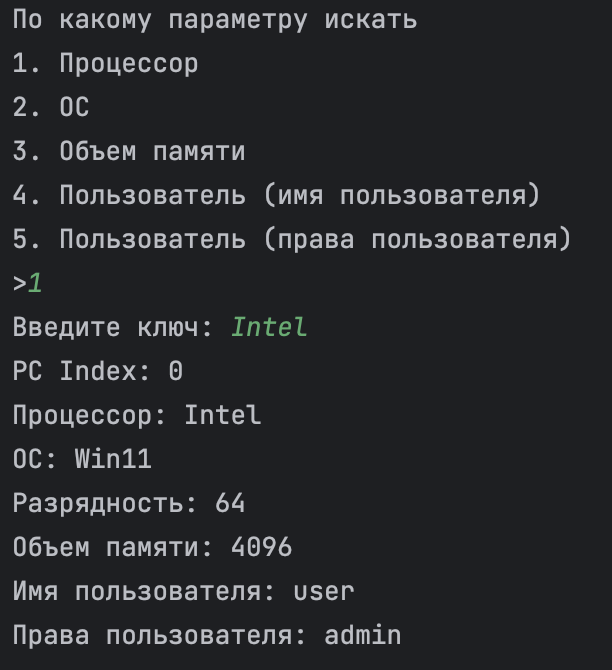
Задача 2.b:



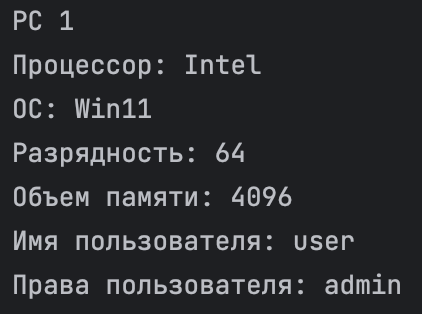
Задача 2.c:



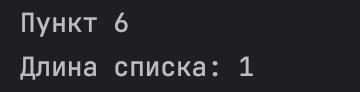
Задача 2.d:



Задача 2.e:



Задача 2.f:



**Заключение**

По результатам практической работы я научился создавать классы с определенными полями, методами, конструкторами, научился работать со списками, добавлять и удалять элементы.