МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ   
Федеральное государственное бюджетное образовательное учреждение высшего образования

**“МОСКОВСКИЙ АВТОМОБИЛЬНО-ДОРОЖНЫЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ (МАДИ)”**

**Факультет автомобильного транспорта**

**Кафедра Высшей математики**

Отчёт по Лабораторной работе №4 на тему:

“Задача о заполнении рюкзака”

По дисциплине: “Структуры и алгоритмы обработки данных”

Выполнил: Жиленко А.А.

Группа 1бПМ

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Принял старший преподаватель,

Кутейников И.А.

Подпись\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2024

Содержание

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

**Цель и постановка задачи**

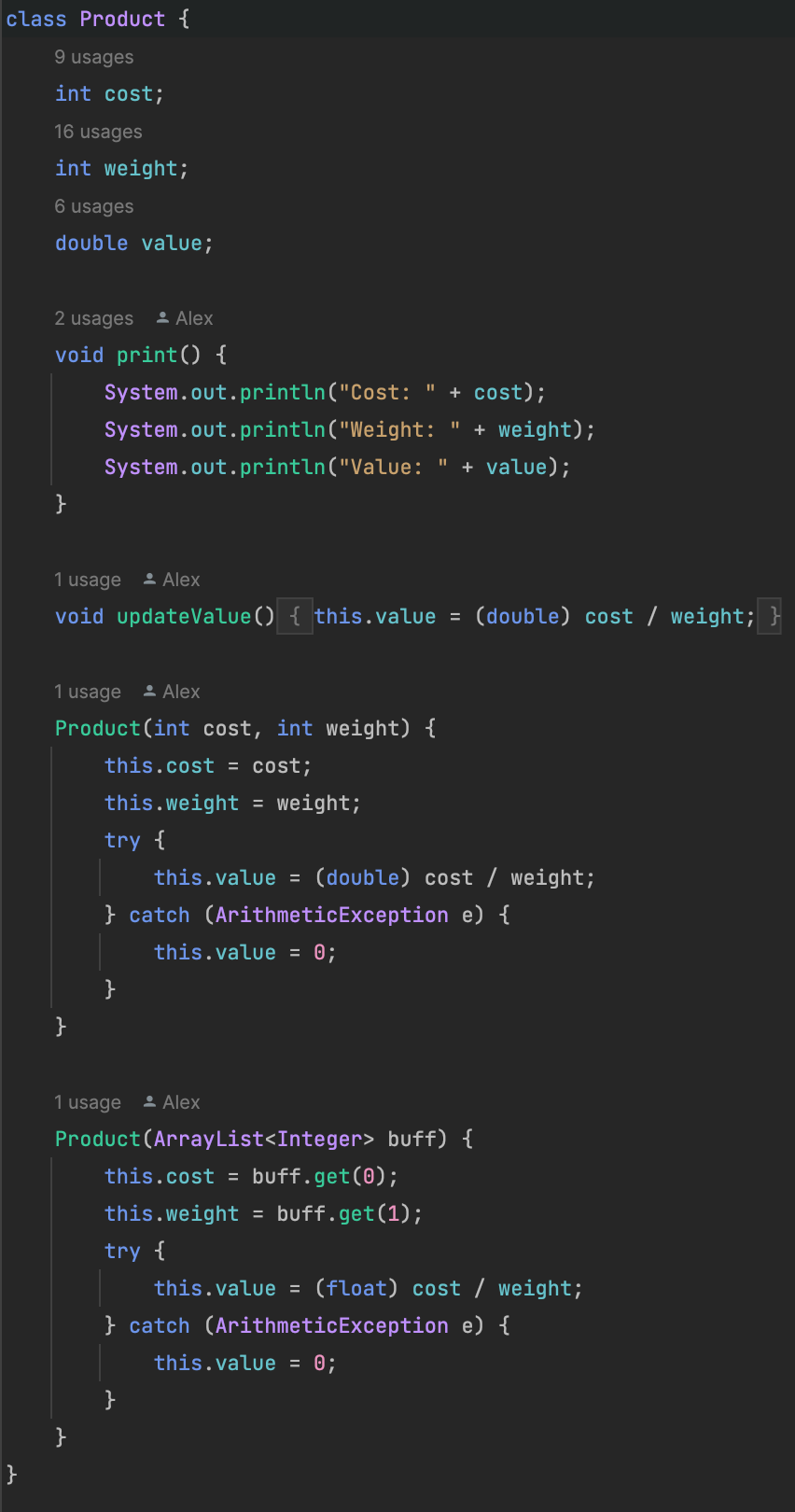
**Цель работы:**  Реализовать программу решающую задачу о заполнении рюкзака: есть набор предметов, каждый предмет имеет название, вес и ценность. Требуется сложить вещи с максимальной стоимостью в рюкзак, имеющий ограничение по весу.

**Задачи:**

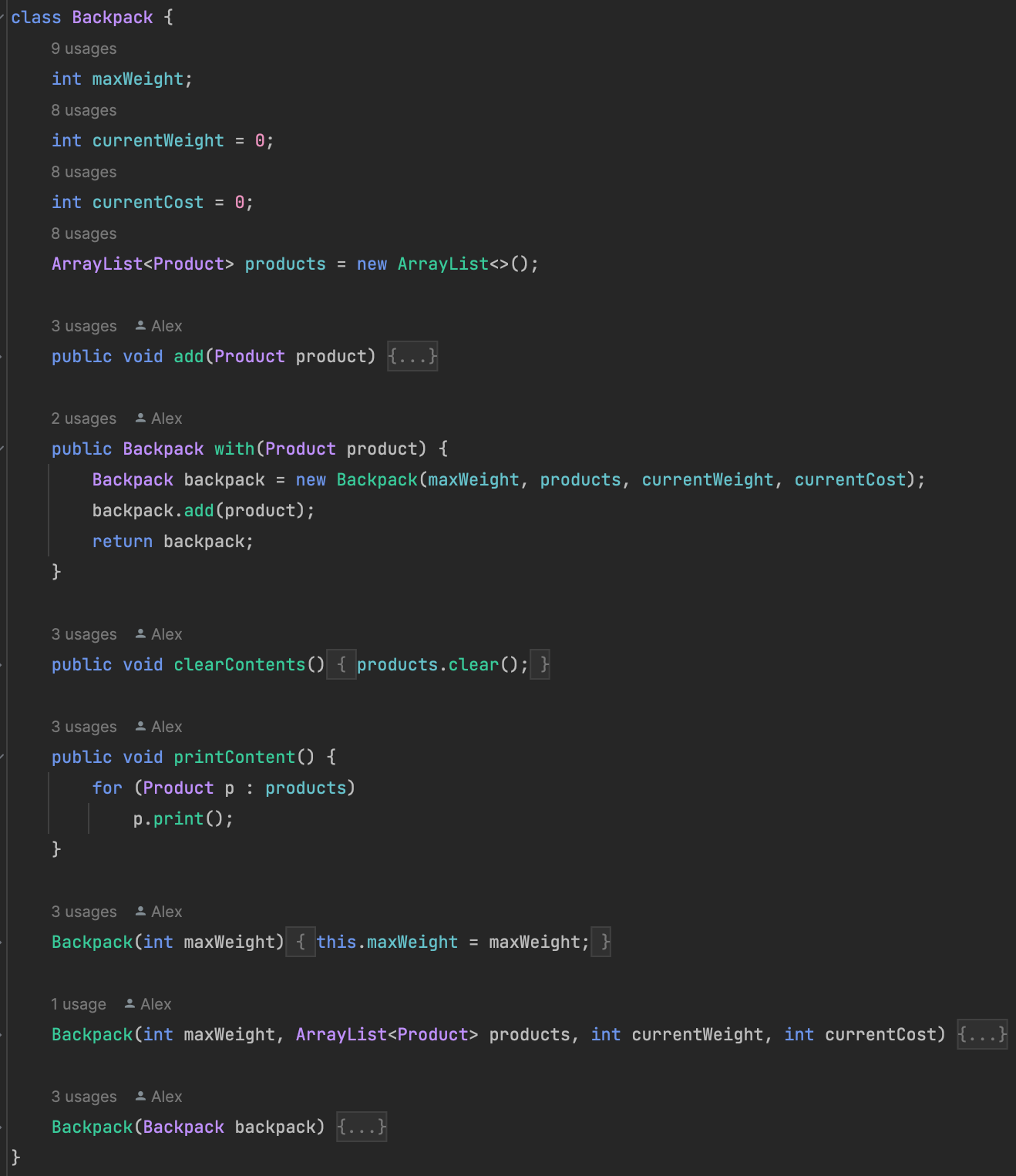
1. В программе должен присутствовать класс «Предмет», обладающий полями: название, вес, цена; и класс «Рюкзак», обладающим полями: максимальный вес, текущий вес, список предметов в рюкзаке.
2. Реализовать методы решения:
   1. Рекурсивный метод
   2. Жадный алгоритм
   3. Метод динамического программирования
3. Реализовать:
4. Заполнение списка из файла
5. Добавление предмета
6. Изменение предмета
7. Удаление предмета
8. Задание максимального веса рюкзака
9. Просмотр содержимого рюкзака
10. Выбор способа решения задачи
11. Сравнение способов решения

**Алгоритм решения**

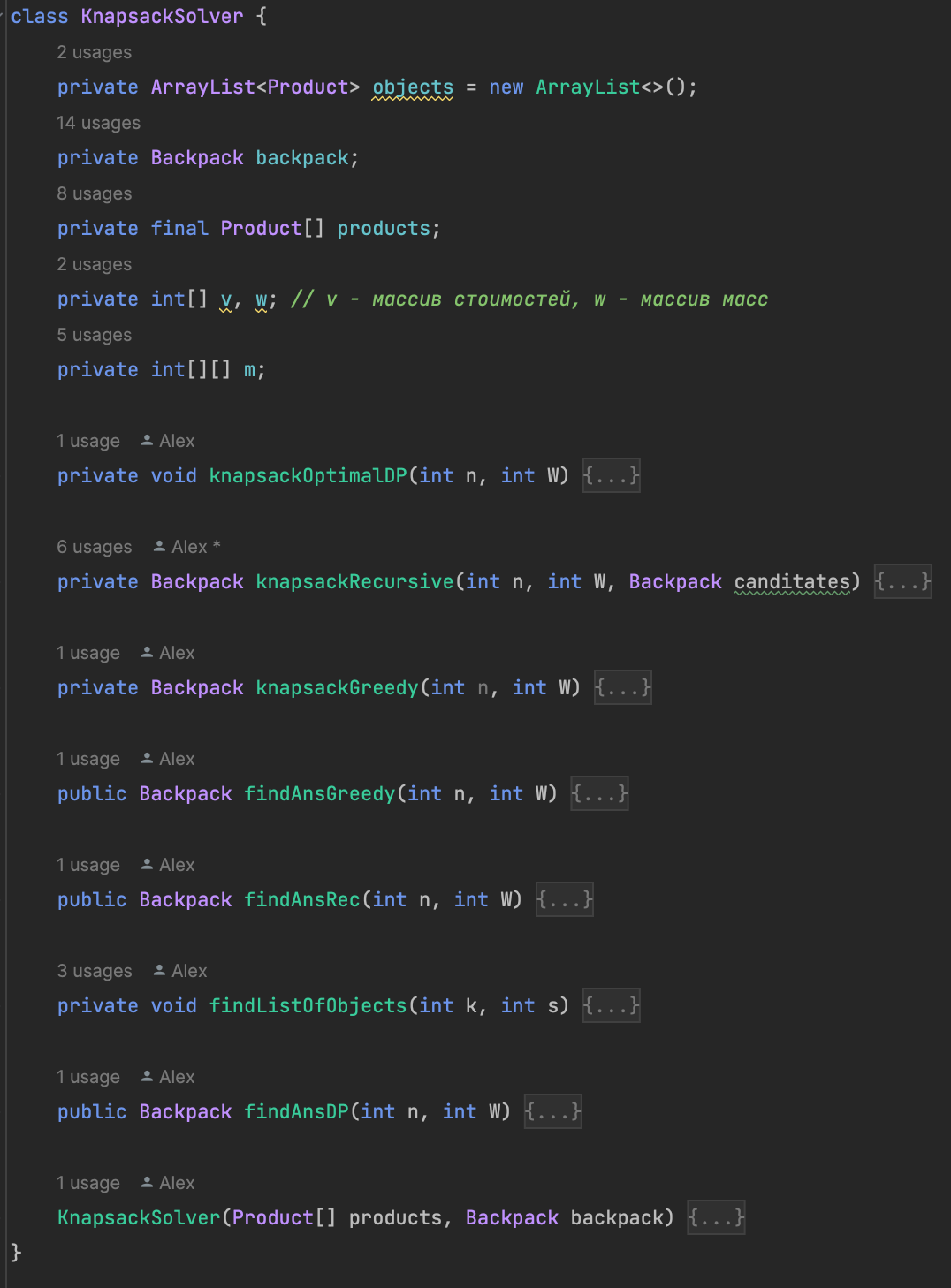
Задача 1:



Класс продукт с полями: стоимость, вес, ценность.

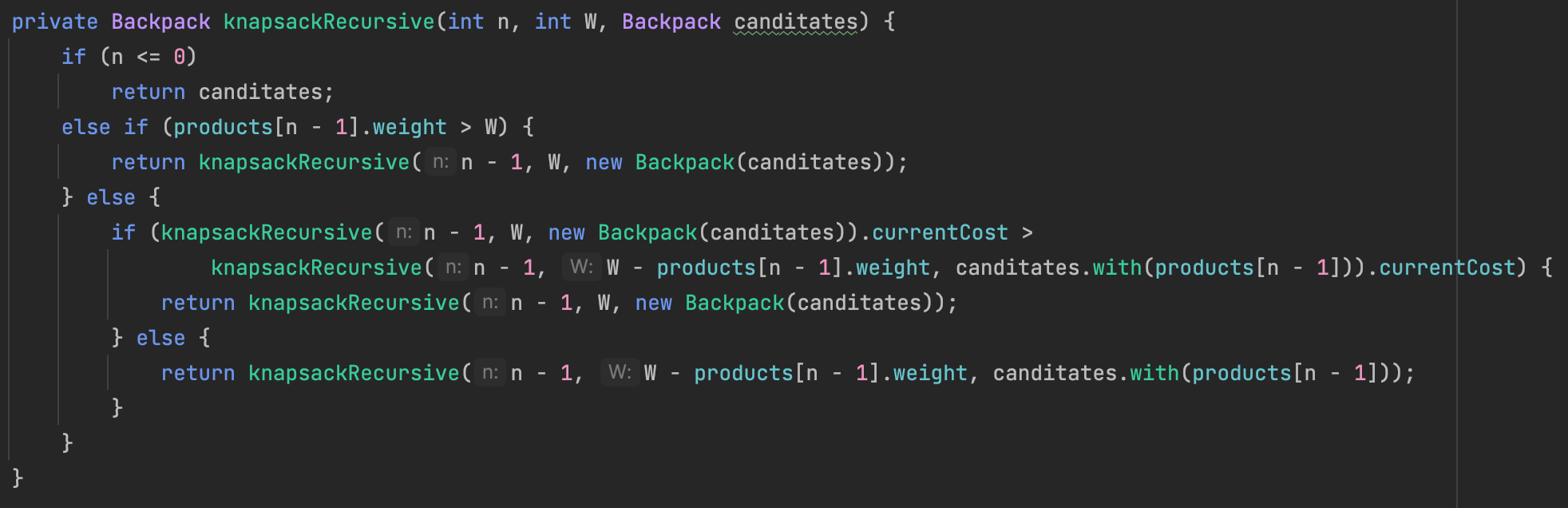


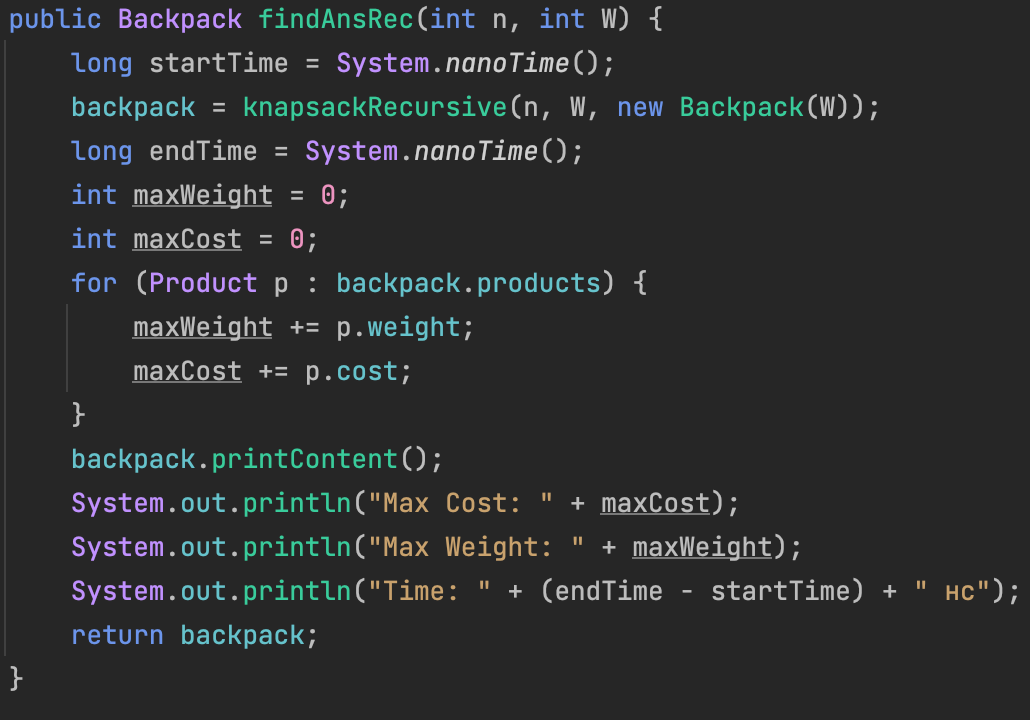
Класс рюкзак с полями: максимальный вес, текущий вес, текущая стоимость, список предметов в рюкзаке.



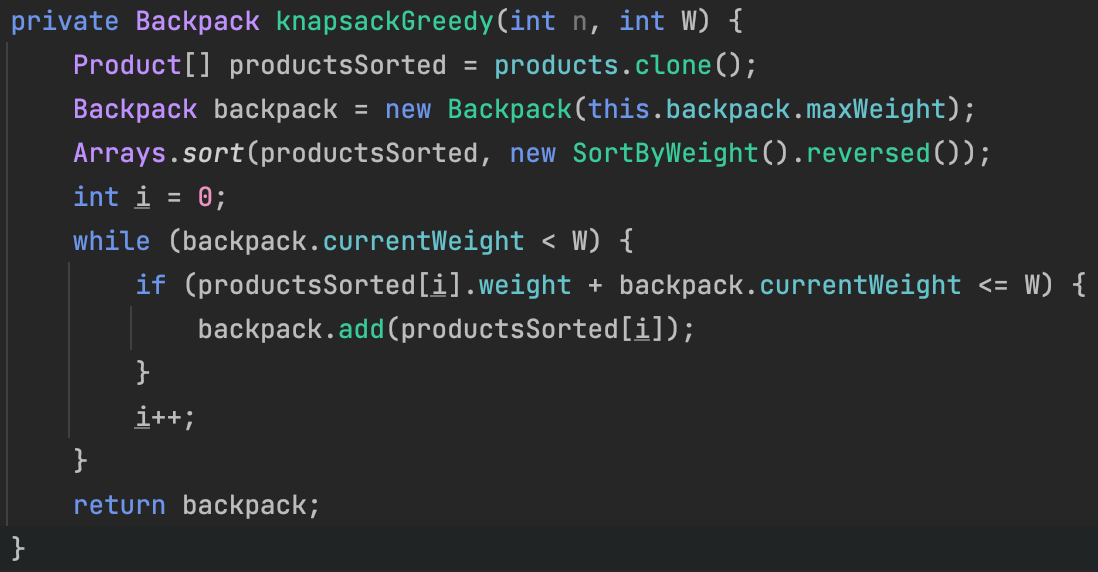
Также был создан класс KnapsackSolver, решатель задачи о рюкзаке. Этот класс содержит методы решения задачи, а также вспомогательные методы для сбора и вывода информации на экран пользователя.

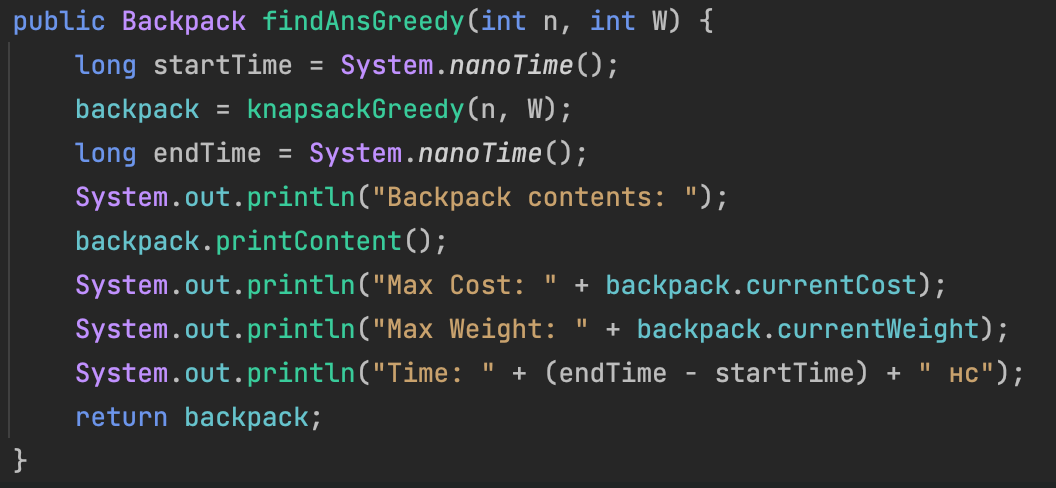
Задача 2.a:





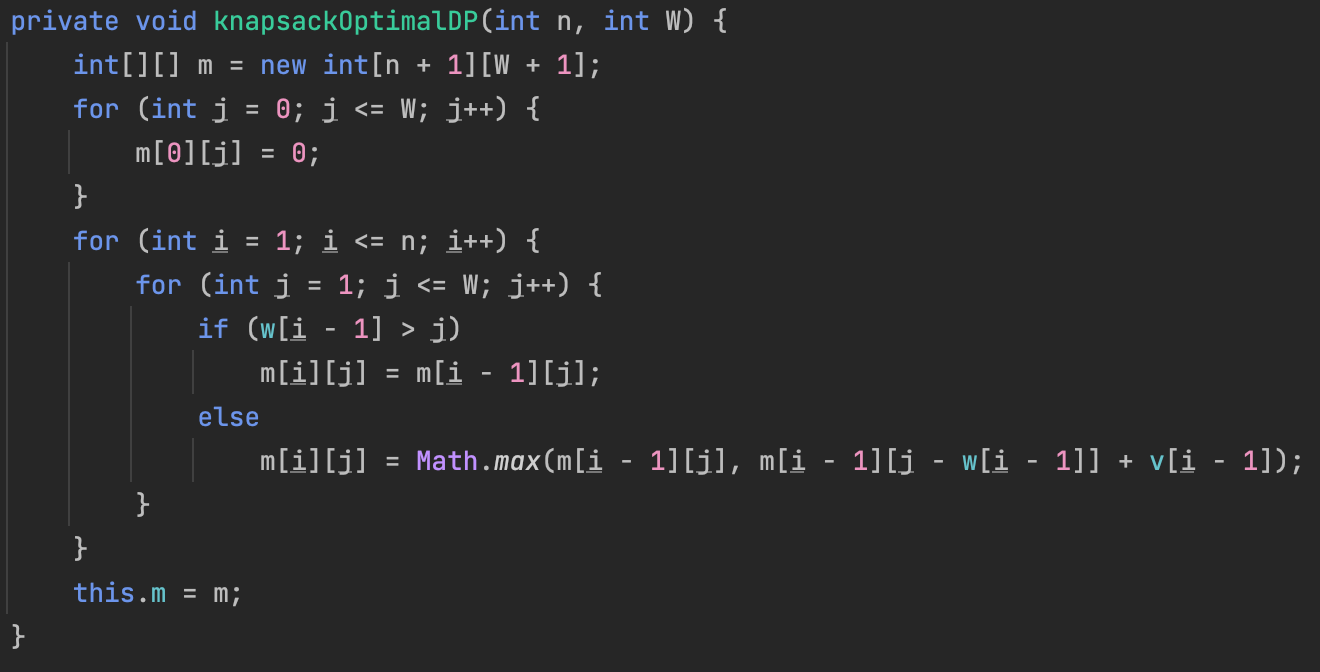
Задача 2.b:





Жадный алгоритм выполняется по стратегии выбора самого тяжелого предмета.

Задача 2.c:



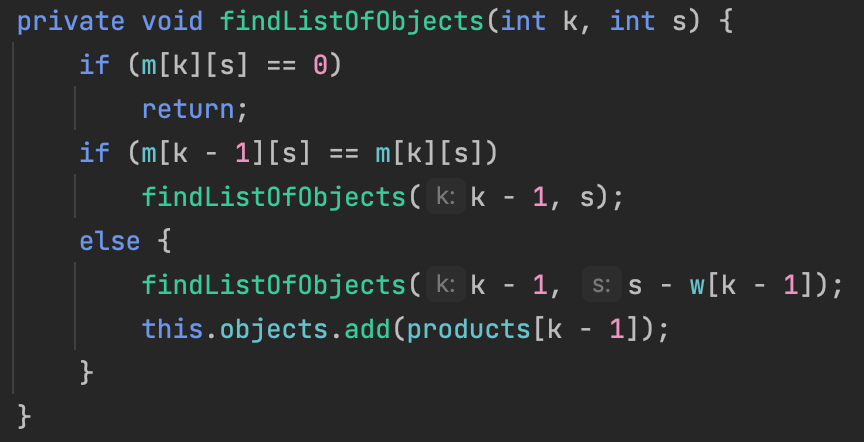
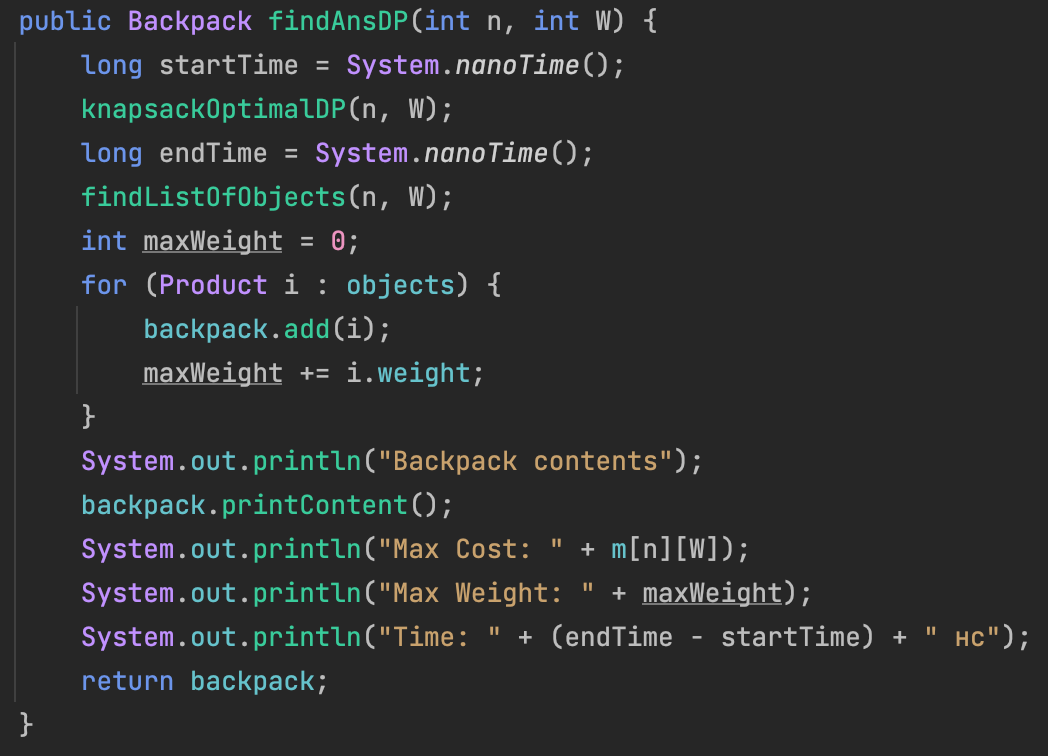
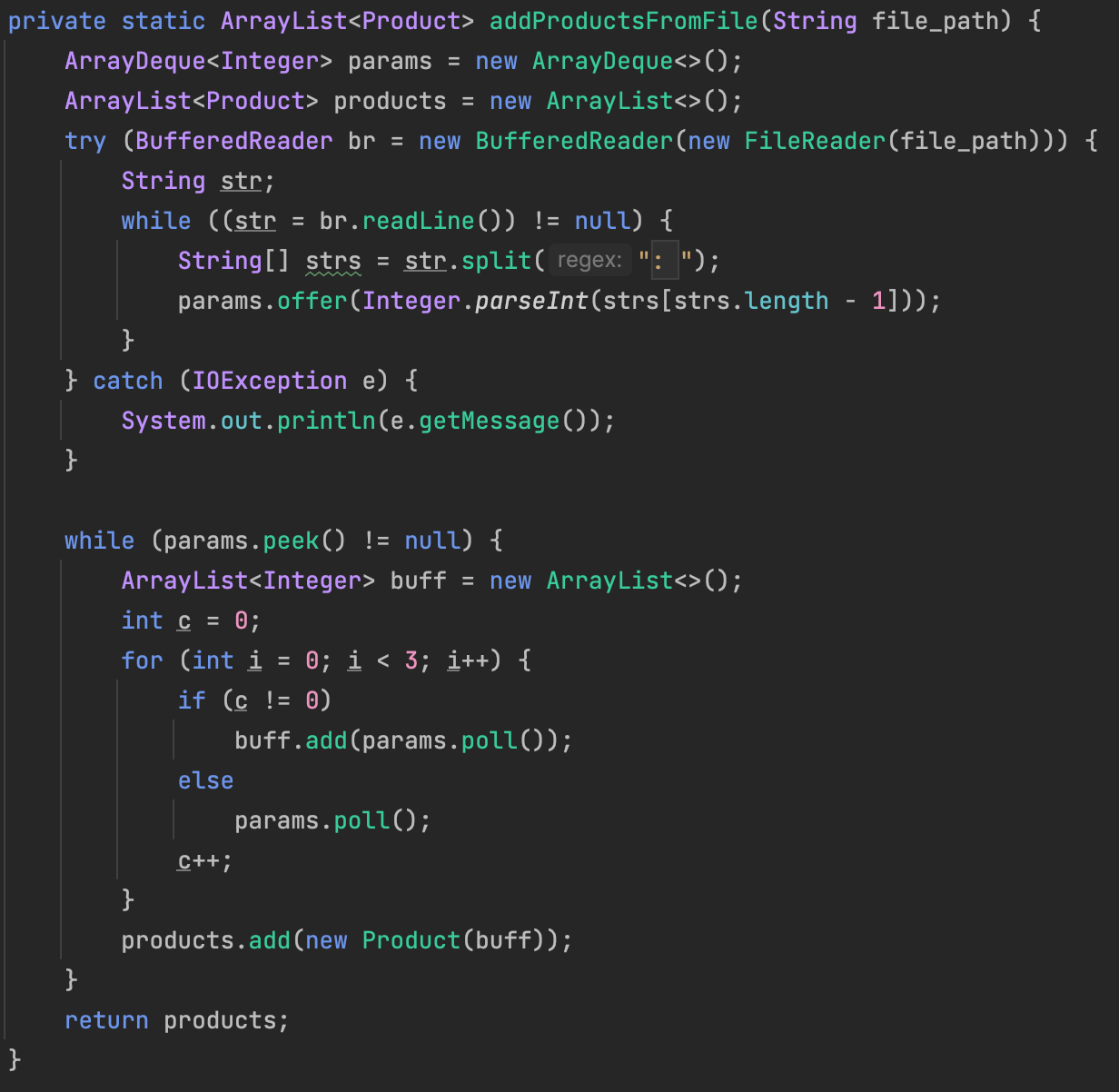


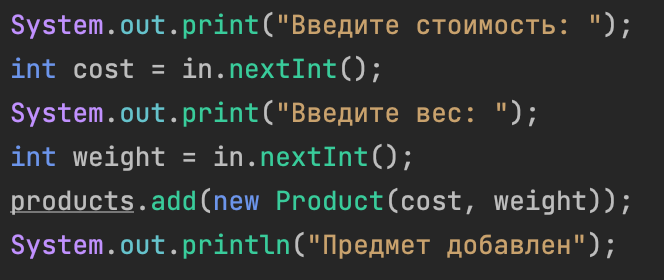
Рисунок 1 Рекурсивный алгоритм нахождения номеров предметов которые находятся в рюкзаке



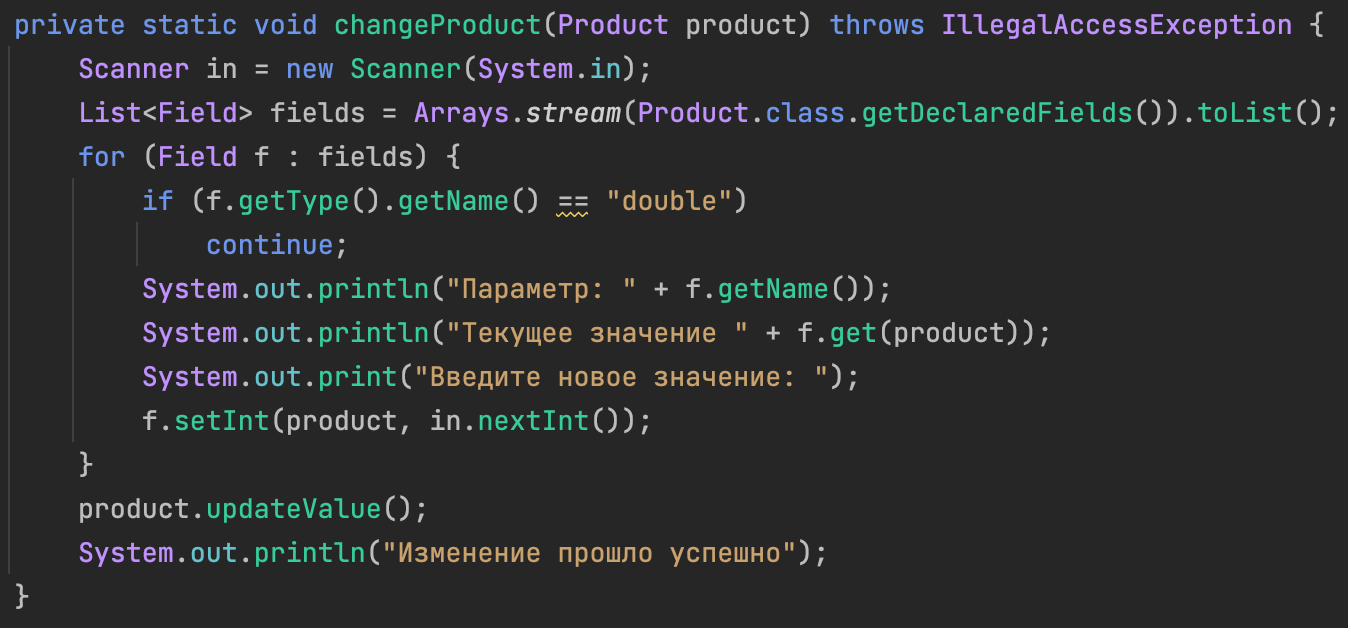
Задача 3.a:



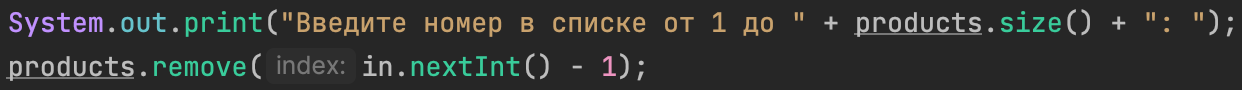
Задача 3.b:



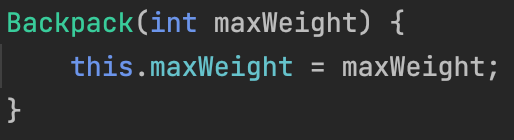
Задача 3.c:



Задача 3.d:

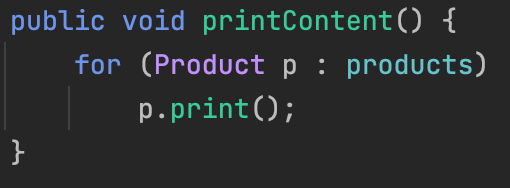


Задача 3.e:

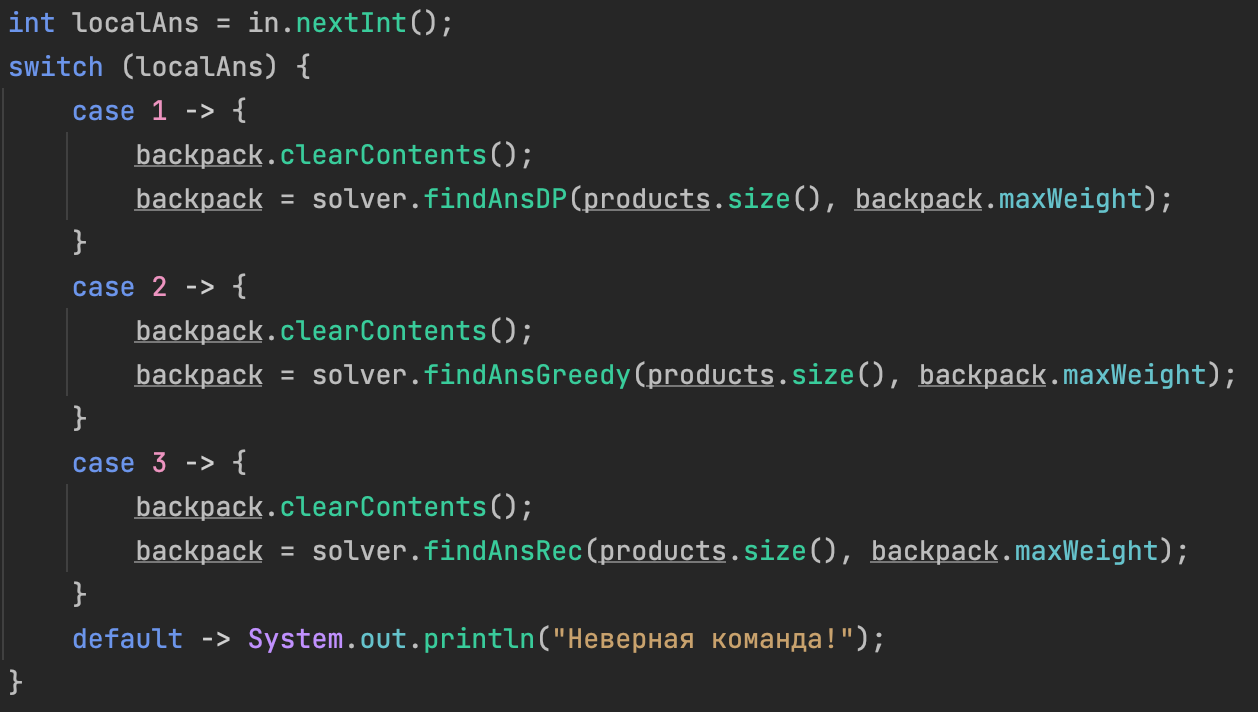


Максимальный вес рюкзака задается при создании рюкзака соответствующим конструктором класса.

Задача 3.f:



Задача 3.g:



Задача 3.h:

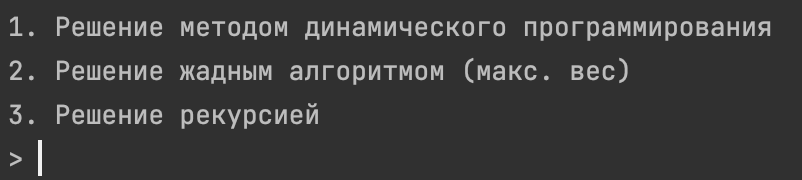
После выполнения каждого алгоритма происходит замер времени его работы и вывод, после окончания выполнения, на экран.

**Результат решения**

Файл хранения информации о предметах вида:

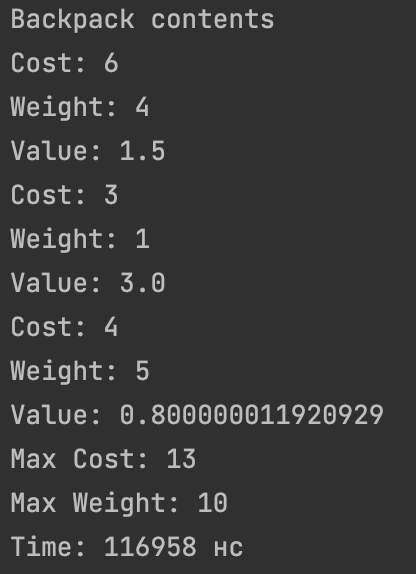
****

У пользователя спрашивается путь к файлу, после чего идет его обработка.

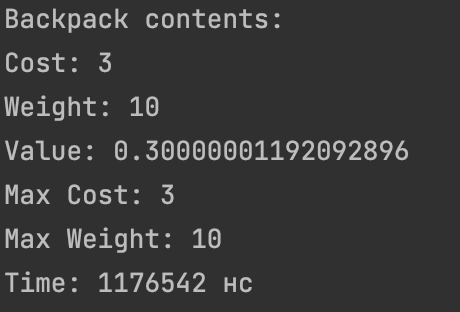


У пользователя спрашивается, каким методом он хочет решить задачу. Вес также задается перед решением, в этом пример он равен 10.

Решение методом динамического программирования:

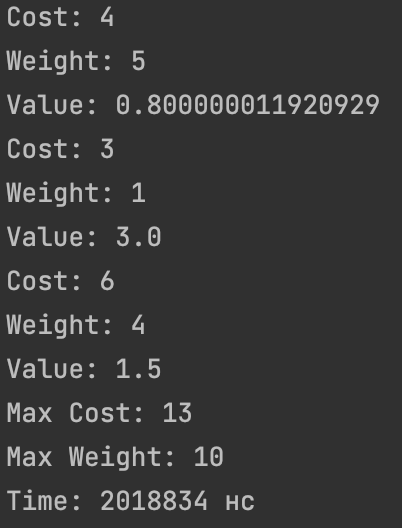


Решение методом жадного алгоритма:



Время выполнения считалось вместе с сортировкой списка предметов. Сложность такого жадного алгоритма получается равной O(nlogn) из-за сложности алгоритма сортировки.

Решение рекурсивным методом:



**Заключение**

По итогам лабораторной работы я познакомился с методом жадного алгоритма, а также сравнил его с методами динамического программирования и рекурсии. Также было реализовано решение задачи о рюкзаке и выполнено сравнение трех методов решения задачи.