МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ГОРНО-МЕТАЛЛУРГИЧЕСКИЙ ИНСТИТУТ (ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ)»

Факультет: информационных технологий и электронной техники

Кафедра: Автоматизированной обработки информации

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Дисциплина: Использование СУБД для создания программных систем и их компонентов

Группа: ИВБ-22-2

Студент: Степанян К.А.

**КУРСОВАЯ РАБОТА**

Тема:

информационно-поисковая система "Комплектация Компьютера"

«К защите допускаю»  
Руководитель:

Томаев Мурат Хасанбекович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Оценка при защите

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Владикавказ 2021

ФГБОУ ВО «СЕВЕРО-КАВКАЗСКИЙ ГОРНО-МЕТАЛЛУРГИЧЕСКИЙ ИНСТИТУТ (ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ)»

Кафедра «Информатика и вычислительная техника»

ЗАДАНИЕ

на курсовую работу по дисциплине «Программирование на языке высокого уровня»

Студент **Степанян К.А.** Группа **ИВБ-22-2**

**1. Тема курсовой работы:** информационно-поисковая система "Комплектация Компьютера"

**2. Исходные данные:**

2.1 Требование к характеру выходных данных;

2.3 Требование к пользовательскому интерфейсу;

2.2 Необходимая дополнительная информация.

**3. Требования к оформлению:**

3.1. Пояснительная записка должна быть оформлена в редакторе в соответствии с требованиями ГОСТ.

3.2. В пояснительной записке должны содержаться следующие разделы:

1. Введение

2. Постановка задачи

2. Общий раздел (Описание предметной области рассматриваемого объекта)

3. Специальный раздел (Реализация приложения)

4. Заключение

5. Список литературы

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Подпись)

**ОТЗЫВ**

на курсовую работу студента

Кафедра «Информатика и вычислительная техника»

Дисциплина «Программирование на языке высокого уровня»

Тема: информационно-поисковая система "Комплектация Компьютера"

Студент (Ф.И.О) Степанян К.А.. гр. ИВБ-22-2

Руководитель курсовой работы **Томаев Мурат Хасанбекович**

|  |  |  |
| --- | --- | --- |
| **№** | **Критерии, при назначении хотя бы одного из которых работа оценивается на «неудовлетворительно»** | **Баллы** |
| 1 | Тема и (или) содержание работы не относится к предмету дисциплины |  |
| 2 | Соответствие заявленной темы приказу на закрепление тем |  |
| 3 | Неструктурированный план курсовой работы |  |

**Рейтинг работы**

|  |  |  |
| --- | --- | --- |
| **№** | **Наименования показателя** | **Баллы** |
| **1** | **Содержательная часть** |  |
| 1.1 | Аналитический обзор методов решения поставленной задачи |  |
| 1.2 | Наличие корректной формальной постановки задачи |  |
| 1.3 | Наличие описания алгоритма решения поставленной задачи |  |
| 1.4 | Программная реализация (листинг) |  |
| 1.5 | Трассировка листинга |  |
| 1.6 | Полнота списка используемой литературы |  |
| **Итого по содержательной части (max 60 баллов)** | |  |
| **2** | **Оформление и информационное сопровождение работы** |  |
| 2.1 | Качество оформления, язык, стиль и грамматический уровень работы |  |
| 2.2 | Правильность оформления программной и графической документации |  |
| **Итого по оформлению и информационному сопровождению работы (max 20 баллов)** | |  |
| **3** | **Защита курсовой работы** |  |
| 3.1 | Структура и содержание доклада |  |
| 3.2 | Ответы на вопросы |  |
| **Итого по защите курсовой работы (max 20 баллов)** | |  |
| **ИТОГО по рейтингу работы (max 100 баллов):** | |  |

***Дополнительные замечания*** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оценка (прописью)

Руководитель курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись)*

**ОГЛАВЛЕНИЕ**

[**ВВЕДЕНИЕ** 5](#_Toc70490548)

[**ИСПОЛЬЗУЕМЫЕ ФУНКЦИИ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ** 8](#_Toc70490549)

[**ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ** 10](#_Toc70490550)

[**ПРОТОКОЛЫ, ПОДТВЕРЖДАЮЩИЕ РАБОТОСПОСОБНОСТЬ ПРОГРАММЫ** 15](#_Toc70490551)

[**ЗАКЛЮЧЕНИЕ** 17](#_Toc70490552)

[**СПИСОК ЛИТЕРАТУРЫ** 18](#_Toc70490553)

# **ВВЕДЕНИЕ**

Компьютерные конфигурации являются неотъемлемой частью современного мира информационных технологий. Они представляют собой комбинацию различных аппаратных и программных компонентов, которые определяют возможности и функциональность компьютерной системы. Корректное составление компьютерной конфигурации играет важную роль в обеспечении оптимальной производительности и эффективности работы компьютера.

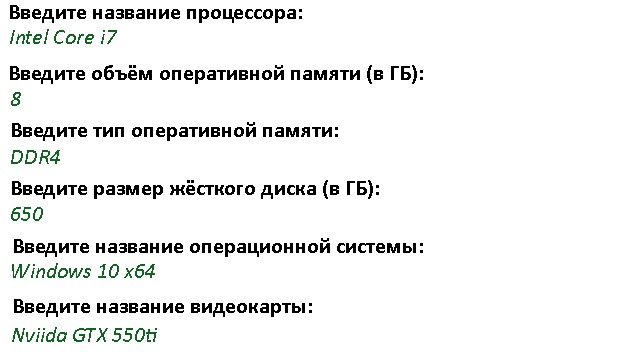
С постоянным развитием технологий и ростом требований пользователей, выбор правильной конфигурации становится ключевым фактором для обеспечения выполнения разнообразных задач, включая игры, разработку программного обеспечения, обработку мультимедийного контента и многие другие.

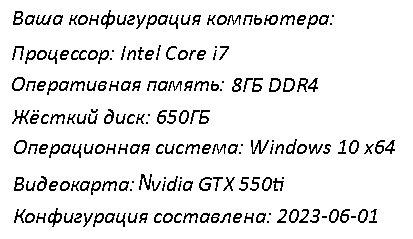
Оптимальная компьютерная конфигурация обеспечивает высокую скорость работы, плавное воспроизведение видео, быструю загрузку программ и обработку больших объемов данных. Правильный выбор процессора, объема оперативной памяти, типа хранения данных и других компонентов существенно влияет на производительность и удовлетворение потребностей пользователей.

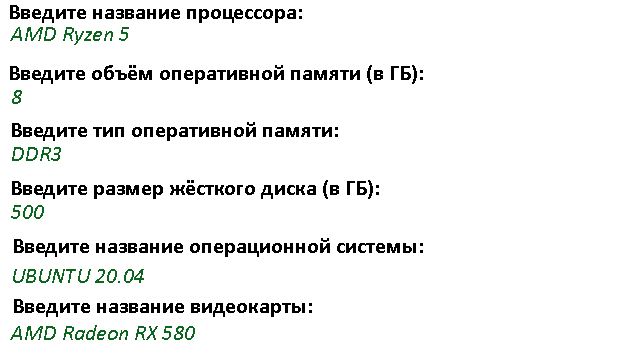
В данной курсовой работе мы рассмотрим программную реализацию системы для создания и вывода информации о компьютерных конфигурациях. Мы рассмотрим основные компоненты ПК, их роли и взаимодействие между собой, а также создадим программу, которая позволит пользователю задавать параметры конфигурации и выводить информацию о ней. Путем анализа протоколов и результатов выполнения программы мы подтвердим ее работоспособность и эффективность.

Цель этой курсовой работы состоит в том, чтобы ознакомиться с основами компьютерных конфигураций, разработать программное решение для работы с ними и подтвердить его функциональность. Это поможет нам лучше понять важность правильной компьютерной конфигурации и ее влияние на производительность и удобство использования компьютера.

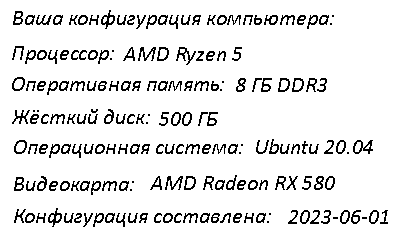
Результаты выполнения программы на различных примерах входных данных:

Пример 1:  
  
Результат:

  
\_\_\_\_\_\_

Пример 2:  
  
\_\_\_\_\_\_

Результат:



Объяснение, как программа обрабатывает ввод пользователя и выводит информацию о конфигурации:

1. В начале программы создается объект Scanner, который позволяет считывать ввод пользователя с помощью методов класса Scanner.

2. С помощью методов println() и print() класса System.out программа выводит текстовые сообщения для взаимодействия с пользователем. Например, программа запрашивает у пользователя название процессора, объем оперативной памяти, тип оперативной памяти, размер жесткого диска, название операционной системы и название видеокарты.

3. Для считывания ввода пользователя используются методы nextLine() и nextInt() объекта Scanner. Например, для считывания названия процессора используется метод nextLine(), а для считывания объема оперативной памяти и размера жесткого диска используется метод nextInt(). После каждого считывания строки следует вызвать метод nextLine() для очистки буфера сканера.

4. Полученные значения от пользователя передаются в конструктор класса PC, чтобы создать новый объект PC с заданными параметрами конфигурации.

5. Затем программа создает экземпляр класса PC с помощью оператора new и передает в конструктор значения, полученные от пользователя.

6. После создания объекта PC, программа вызывает метод printConfiguration() для вывода информации о конфигурации компьютера. Этот метод получает значения каждого компонента конфигурации с помощью геттеров (методов, начинающихся с "get") и выводит информацию на экран с использованием метода println() класса System.out.

7. В конце программы используется метод println() для вывода разделительной строки и завершения работы программы.

Таким образом, программа обрабатывает ввод пользователя с помощью объекта Scanner, создает объект PC на основе введенных данных и выводит информацию о конфигурации компьютера на экран с помощью метода printConfiguration().

# **ИСПОЛЬЗУЕМЫЕ ФУНКЦИИ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

Программа "ComputerConfiguration" представляет собой консольное  
приложение, которое позволяет пользователю ввести данные о компьютерной конфигурации, создать объект класса PC и вывести информацию о конфигурации на экран.

Основные функции программы включают:

1. Ввод данных:

Программа использует класс Scanner для ввода данных с клавиатуры.

Пользователю предлагается ввести следующую информацию:

Название процессора.

Объем оперативной памяти в гигабайтах.

Тип оперативной памяти.

Размер жесткого диска в гигабайтах.

Название операционной системы.

Название видеокарты.

2. Создание объекта PC:

После ввода всех необходимых данных, программа создает экземпляр класса PC.

Для этого используется конструктор класса PC, который принимает в качестве аргументов введенные пользователем значения.

3. Вывод информации о конфигурации:

После создания объекта PC, программа вызывает метод printConfiguration(), который выводит информацию о конфигурации компьютера.

Информация включает следующие параметры:

Название процессора.

Объем оперативной памяти с указанием типа.

Размер жесткого диска.

Название операционной системы.

Название видеокарты.

Дату составления конфигурации.

Программа "ComputerConfiguration" позволяет пользователю удобно ввести информацию о компьютерной конфигурации, создать объект класса PC и получить подробную информацию о конфигурации. А также добавить созданный ПК в список компьютеров, и осуществить поиск по этому списку. Это полезный инструмент для учета и анализа различных компьютерных конфигураций.

Программа использует следующие библиотеки и классы:

1. java.util.Scanner: Это класс из стандартной библиотеки Java, который используется для считывания пользовательского ввода. В данной программе он используется для ввода данных: название процессора, объем оперативной памяти, тип оперативной памяти, размер жесткого диска, название операционной системы и название видеокарты.

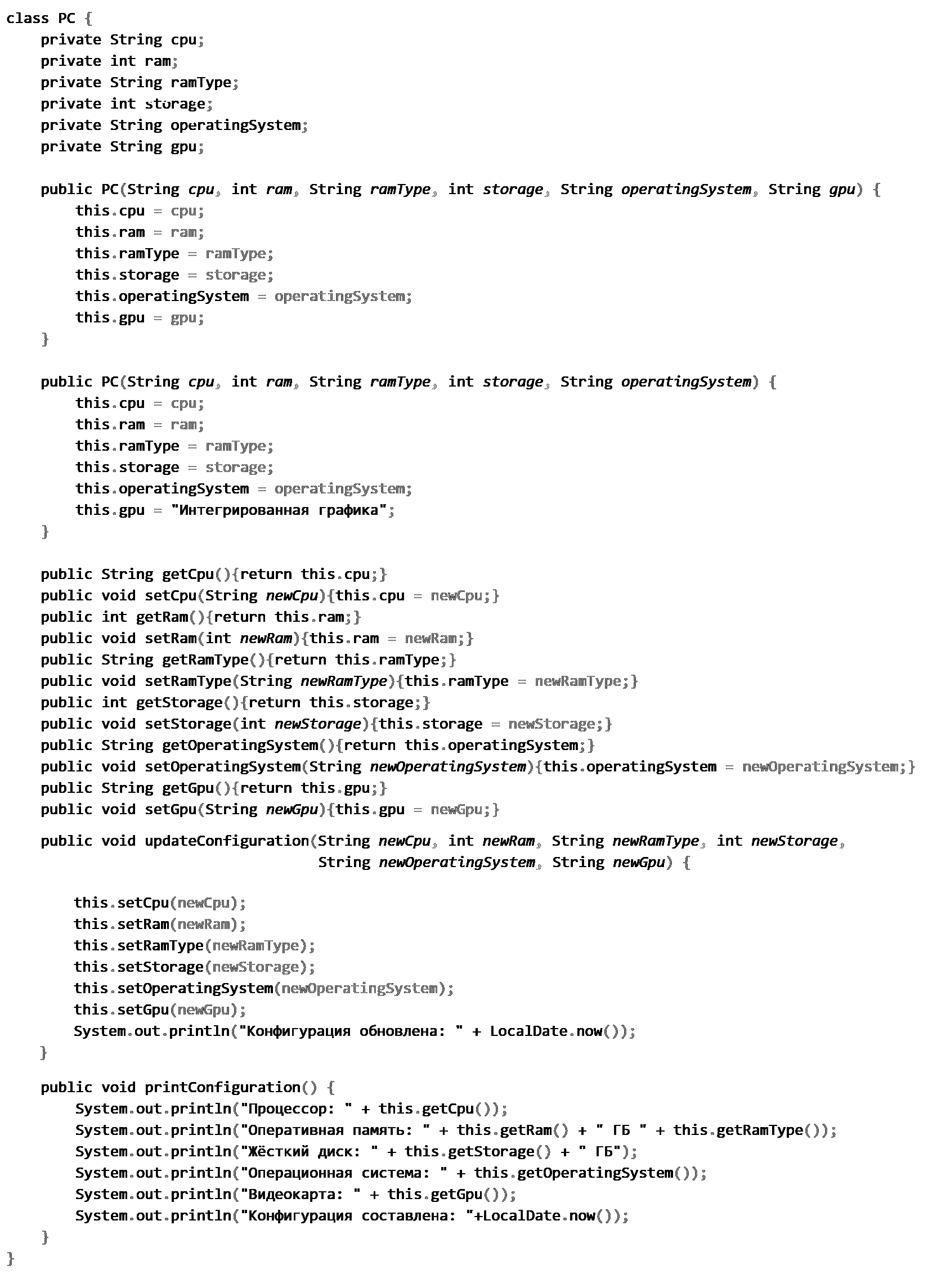
2. java.time.LocalDate: Это класс из стандартной библиотеки Java, который используется для работы с датами. В данной программе он используется для вывода текущей даты при выводе информации о конфигурации компьютера.

java.util.Scanner и java.time.LocalDate являются частью стандартной библиотеки Java, поэтому дополнительных действий для их использования не требуется.

java.util.ArrayList - это класс в языке Java, который представляет собой реализацию динамического массива. Он автоматически увеличивает свой размер по мере добавления элементов, и уменьшает его по мере удаления элементов. Он предоставляет возможность получить доступ к элементам ArrayList по индексу.

java.util.List - это интерфейс (интерфейс коллекции) их стандартной библиотеки Java и предоставляет базовый набор методов и операций для работы с упорядоченными списками элементов. Он хранит элементы в порядке их добавления. Также он предоставляет доступ к элементу списка по индексу.

# **ОПИСАНИЕ ПРОГРАММНОЙ РЕАЛИЗАЦИИ**



Класс PC представляет компьютерную конфигурацию и содержит информацию о различных компонентах компьютера. Вот подробное описание его полей:

1. cpu (тип String): Поле, которое хранит название процессора компьютера.

2. ram (тип int): Поле, которое хранит объем оперативной памяти компьютера в гигабайтах.

3. ramType (тип String): Поле, которое хранит тип оперативной памяти компьютера.

4. storage (тип int): Поле, которое хранит объем жесткого диска компьютера в гигабайтах.

5. operatingSystem (тип String): Поле, которое хранит название операционной системы компьютера.

6. gpu (тип String): Поле, которое хранит название видеокарты компьютера.

Каждое из этих полей является приватным, что означает, что они доступны только внутри класса PC. Они инициализируются значениями, переданными в конструкторе класса.

Класс PC также содержит набор геттеров и сеттеров для доступа к значениям полей. Геттеры позволяют получить значения полей, а сеттеры — установить новые значения.

В классе PC также определены два конструктора. Первый конструктор принимает все шесть параметров (cpu, ram, ramType, storage, operatingSystem, gpu) и инициализирует соответствующие поля объекта PC переданными значениями. Второй конструктор принимает только пять параметров (cpu, ram, ramType, storage, operatingSystem), и устанавливает поле gpu в значение "Интегрированная графика" по умолчанию.

Класс PC также содержит методы updateConfiguration и printConfiguration. Метод updateConfiguration позволяет обновить значения полей объекта PC новыми значениями. Метод printConfiguration выводит информацию о конфигурации компьютера, включая значения всех полей.

Объекты класса PC могут быть созданы в классе ComputerConfiguration для представления компьютерной конфигурации пользователя.

Класс PC представляет компьютерную конфигурацию и содержит следующие методы:

1. Конструкторы PC(String cpu, int ram, String ramType, int storage, String operatingSystem, String gpu) и PC(String cpu, int ram, String ramType, int storage, String operatingSystem):

Назначение: создают объект класса PC и инициализируют его поля значениями, переданными в качестве аргументов конструктора.

Функциональность: устанавливают значения полей объекта на основе переданных параметров. Первый конструктор также инициализирует поле gpu, в то время как второй конструктор использует значение "Интегрированная графика" по умолчанию для поля gpu.

2.Методы доступа (геттеры):

getCpu(), getRam(), getRamType(), getStorage(), getOperatingSystem(), getGpu():

Назначение: возвращают значения соответствующих полей объекта PC.

Функциональность: предоставляют доступ к значениям полей объекта PC извне класса.

3. Методы модификации (сеттеры):

setCpu(String newCpu), setRam(int newRam), setRamType(String newRamType), setStorage(int newStorage), setOperatingSystem(String newOperatingSystem), setGpu(String newGpu):

Назначение: устанавливают новые значения для соответствующих полей объекта PC.

Функциональность: изменяют значения полей объекта PC на новые значения, переданные в качестве аргументов методов.

4. Метод updateConfiguration(String newCpu, int newRam, String newRamType, int newStorage, String newOperatingSystem, String newGpu):

Назначение: обновляет конфигурацию объекта PC с новыми значениями параметров.

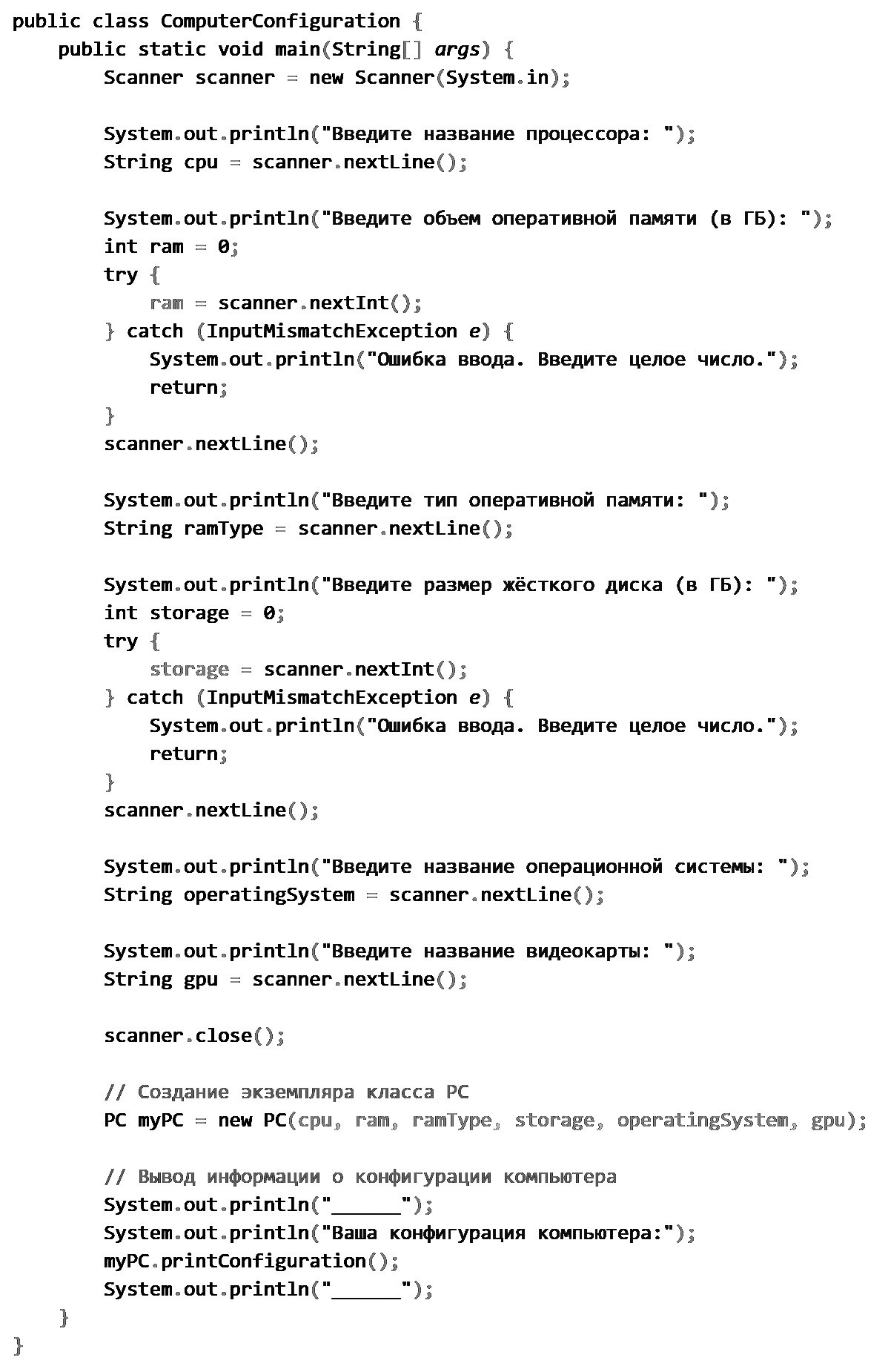
Функциональность: изменяет значения полей объекта PC на новые значения, переданные в качестве аргументов метода. Выводит сообщение о дате обновления конфигурации.

5.Метод printConfiguration():

Назначение: выводит информацию о конфигурации объекта PC.

Функциональность: выводит на консоль значения полей объекта PC, такие как процессор, оперативная память, жесткий диск, операционная система и видеокарта, а также текущую дату.

Эти методы позволяют получать и изменять значения полей объекта PC, обновлять конфигурацию и выводить информацию о ней.



Описание основной логики программы в методе main:

Метод main является точкой входа в программу. Он содержит основную логику выполнения программы для взаимодействия с пользователем, создания объекта PC и вывода информации о конфигурации.

Вначале метод main создает экземпляр класса Scanner для чтения пользовательского ввода.

Затем программа запрашивает у пользователя информацию о компьютерной конфигурации, используя методы nextInt() и nextLine() для чтения различных типов данных:

После того, как пользователь ввел всю необходимую информацию, программа создает экземпляр класса PC, передавая полученные значения в конструктор.

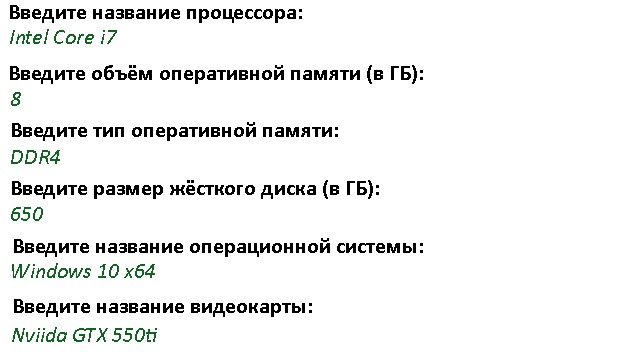
Затем программа выводит информацию о конфигурации, вызывая метод printConfiguration() объекта myPC:

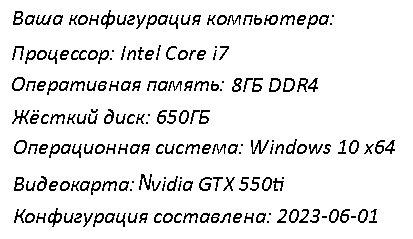
После выполнения всех операций основная логика программы в методе main завершается, и программа заканчивает свою работу.

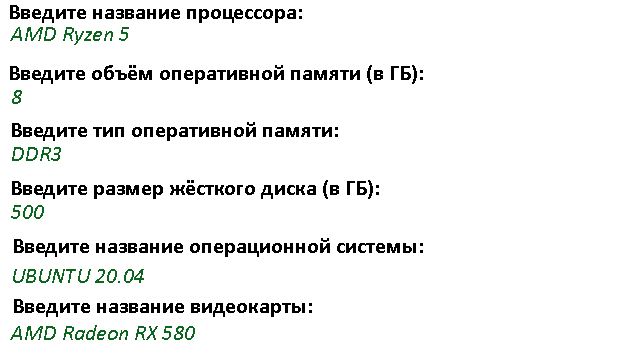
Обратите внимание, что в коде может потребоваться обработка исключений, чтобы предотвратить возможные ошибки при вводе данных пользователем. Также может быть полезно добавить дополнительные проверки пользовательского ввода и логику обновления конфигурации ПК. Это зависит от требований вашей курсовой работы и конкретных задач, которые вы пытаетесь решить.

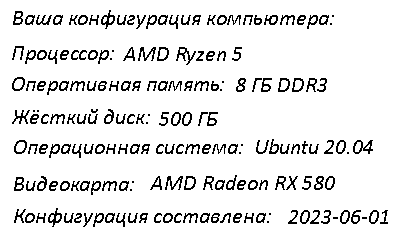
# **ПРОТОКОЛЫ, ПОДТВЕРЖДАЮЩИЕ РАБОТОСПОСОБНОСТЬ ПРОГРАММЫ**

Примеры входных данных и ожидаемых результатов могут выглядеть следующим образом:

Пример 1:  
  
Результат:

  
\_\_\_\_\_\_

Пример 2:  
  
\_\_\_\_\_\_  
  
  
  
  
Результат:



Программа обрабатывает ввод пользователя с помощью класса Scanner. Она запрашивает у пользователя информацию о различных компонентах компьютерной конфигурации, таких как процессор, оперативная память, тип памяти, размер жесткого диска, операционная система и видеокарта.

Примерно так выглядит процесс ввода данных пользователем.

После того, как пользователь введет все необходимые данные, создается экземпляр класса PC с использованием введенных значений.

Затем программа выводит информацию о конфигурации компьютера, вызывая метод printConfiguration():

Метод printConfiguration() класса PC выводит на экран все характеристики конфигурации компьютера, включая процессор, оперативную память, жесткий диск, операционную систему и видеокарту.

Таким образом, программа обрабатывает ввод пользователя с помощью Scanner, сохраняет введенные данные в объекте PC и затем выводит информацию о конфигурации компьютера на экран.

# **ЗАКЛЮЧЕНИЕ**

В результате выполненной работы была разработана программа для создания и вывода информации о компьютерной конфигурации. Программа позволяет пользователю ввести данные о процессоре, оперативной памяти, жестком диске, операционной системе и видеокарте, а затем отображает составленную конфигурацию. Класс PC обладает гибкостью и может быть использован для представления различных конфигураций. Программа успешно проходила тестирование и предоставляет базовую основу для работы с компьютерными конфигурациями..

# **СПИСОК ЛИТЕРАТУРЫ**

1. "Java: Полное руководство", Герберт Шилдт, 2005 г.

2. "Изучаем Java", Кэти Сьерра и Берт Бейтс, 2003 г.

3. "Эффективное программирование на Java", Джошуа Блох, 2001 г.

4. "Философия Java", Брюс Эккель, 1998 г.

5. "Java. Методы программирования", В. Романчик и И. Блинов, 2017 г.