**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена трудового Красного Знамени федеральное государственное бюджетное**

**образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра Математическая кибернетика и информационные технологии

Отчет по лабораторной работе №2.

Выполнил: студент группы \_\_\_БВТ2402\_\_

Стадник Андрей Романович

Москва, 2025

**Цель работы:** освоить основы объектно-ориентированного программирования на языке Java, научиться создавать абстрактные и дочерние классы, конструкторы и методы, использовать принципы наследования, полиморфизма и инкапсуляции, а также закрепить навыки работы со статическими переменными и выводом информации о объектах.

**Задание 1** Создайте иерархию классов в соответствии с вариантом. Ваша иерархия должна содержать:

• абстрактный класс;

• два уровня наследуемых классов (классы должны содержать в себе минимум 3 поля и 2 метода, описывающих поведение объекта);

• демонстрацию реализации всех принципов ООП;

• наличие конструкторов (в том числе по умолчанию);

• наличие геттеров и сеттеров;

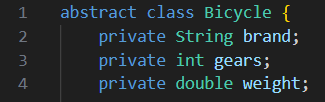
• ввод/вывод информации о создаваемых объектах;

• предусмотрите в одном из классов создание счетчика созданных объектов с использованием статической переменной, продемонстрируйте работу.

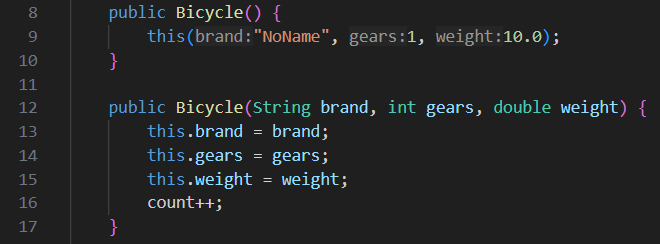
**5 Вариант**. Базовый класс: Велосипед. Дочерние классы: Горный велосипед, Детский велосипед, BMX.

**I) Создание абстрактного класса Bicycle.**

1) В абстрактном классе Bicycle объявлены три основных поля, которые характерны для любого велосипеда:



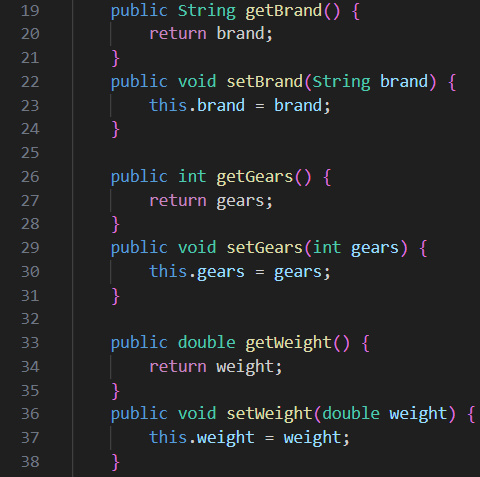
2) Были созданы два конструктора: конструктор по умолчанию — задаёт значения по стандарту; конструктор с параметрами — позволяет при создании объекта сразу задать характеристики.

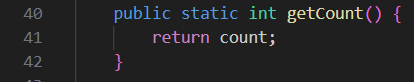


3) Было объявлено два абстрактные метода: ride – “ехать”, brake – “тормозить”. В подклассах эти методы будут реализованы.

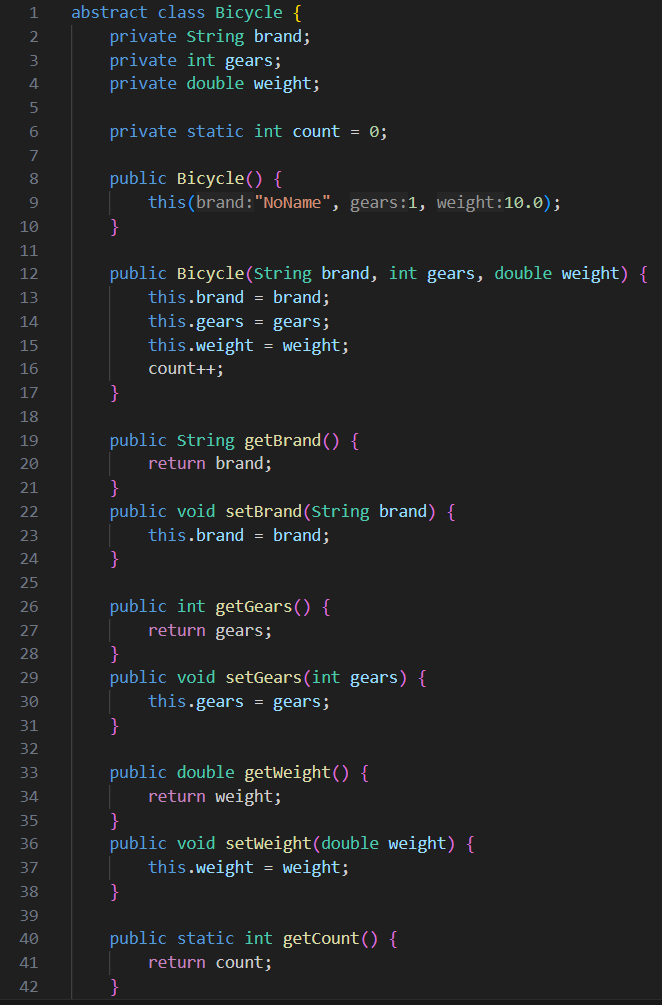


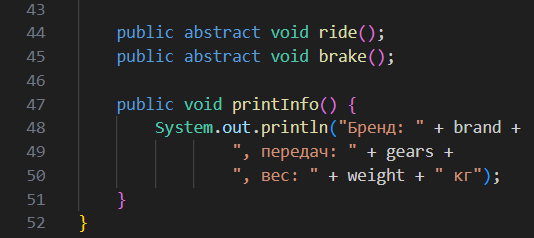
4) Были созданы геттеры и сеттеры для инкапсуляции.



5) Чтобы узнать, сколько велосипедов создано, введена статическая переменная count. Она увеличивается при каждом вызове конструктора.

**Полный код Bycicle:**



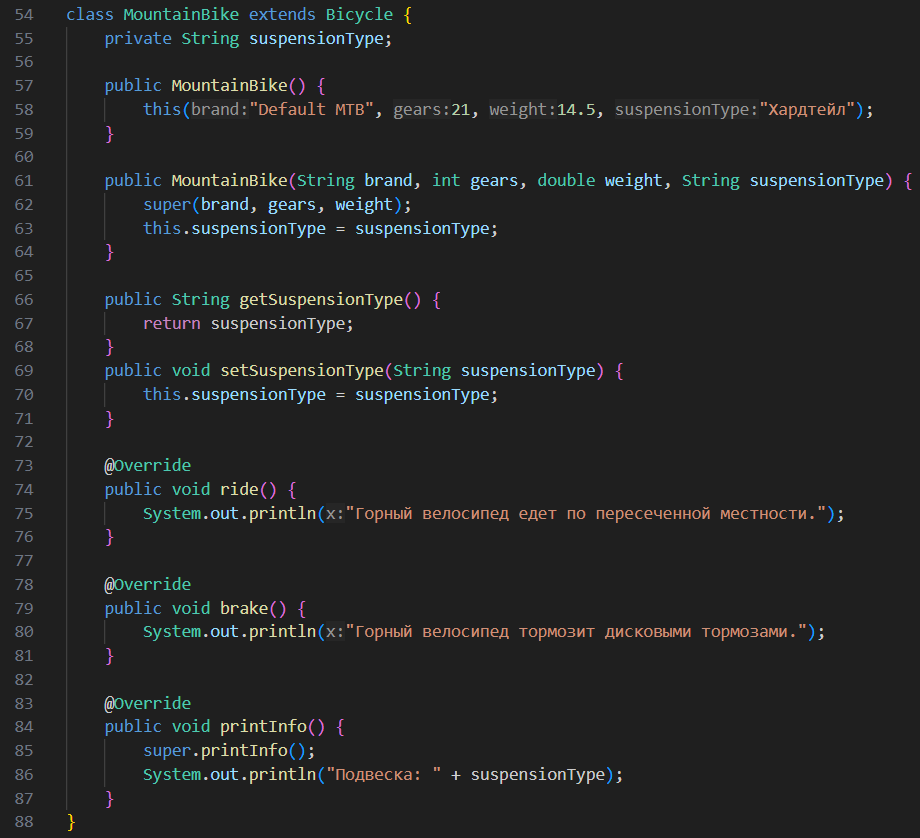


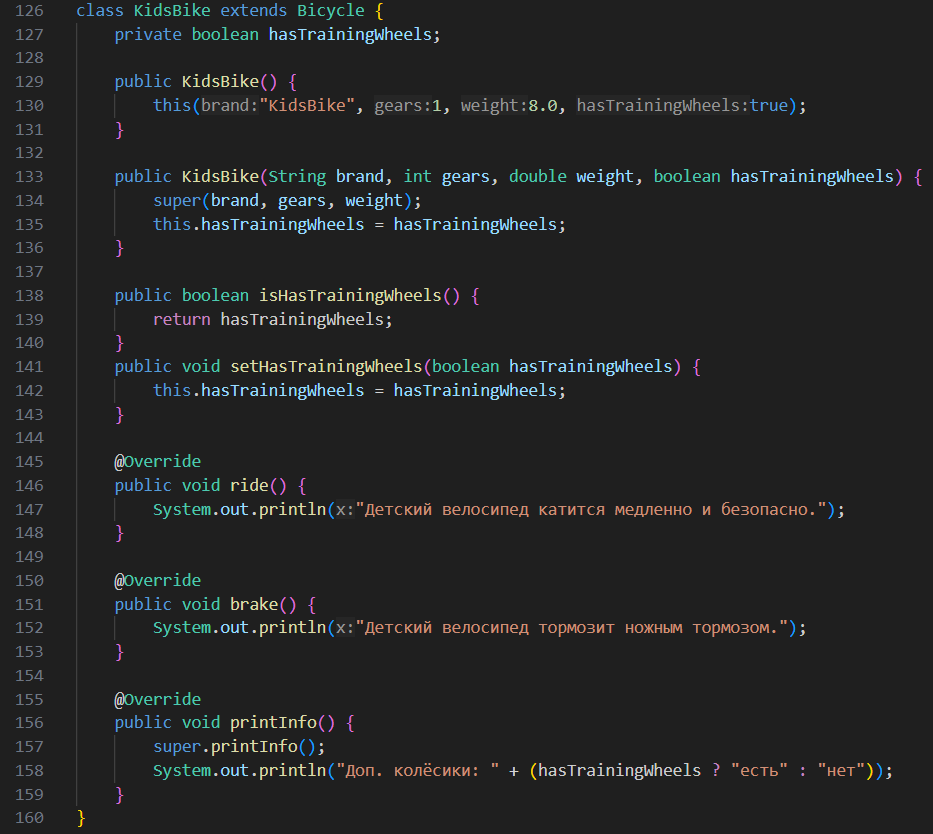
**II) Реализация дочерних классов.**

В соответствии с заданием созданы классы-наследники:

* MountainBike (Горный велосипед) — добавлено поле suspensionType.
* KidsBike (Детский велосипед) — добавлено поле hasTrainingWheels.
* BMX — добавлено поле frameMaterial.

Каждый класс реализует методы ride() и brake(), а также переопределяет printInfo() для вывода полной информации.

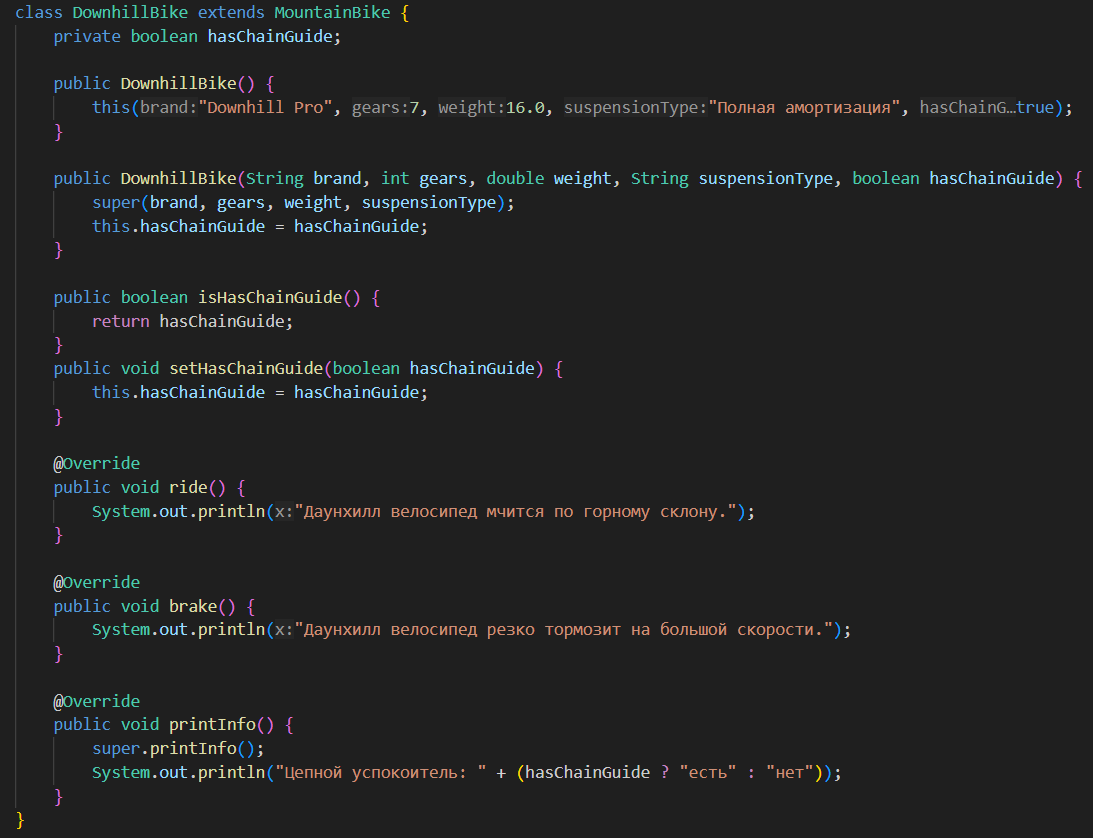


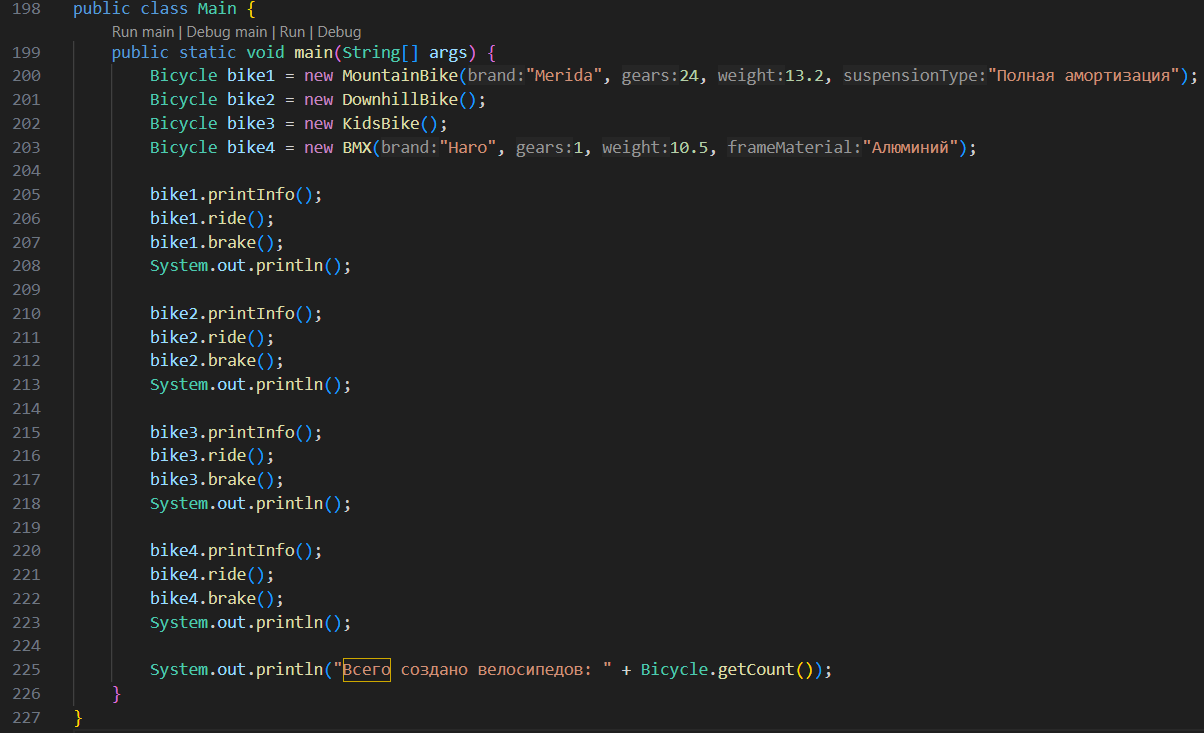
**III) Добавление второго уровня наследования.**

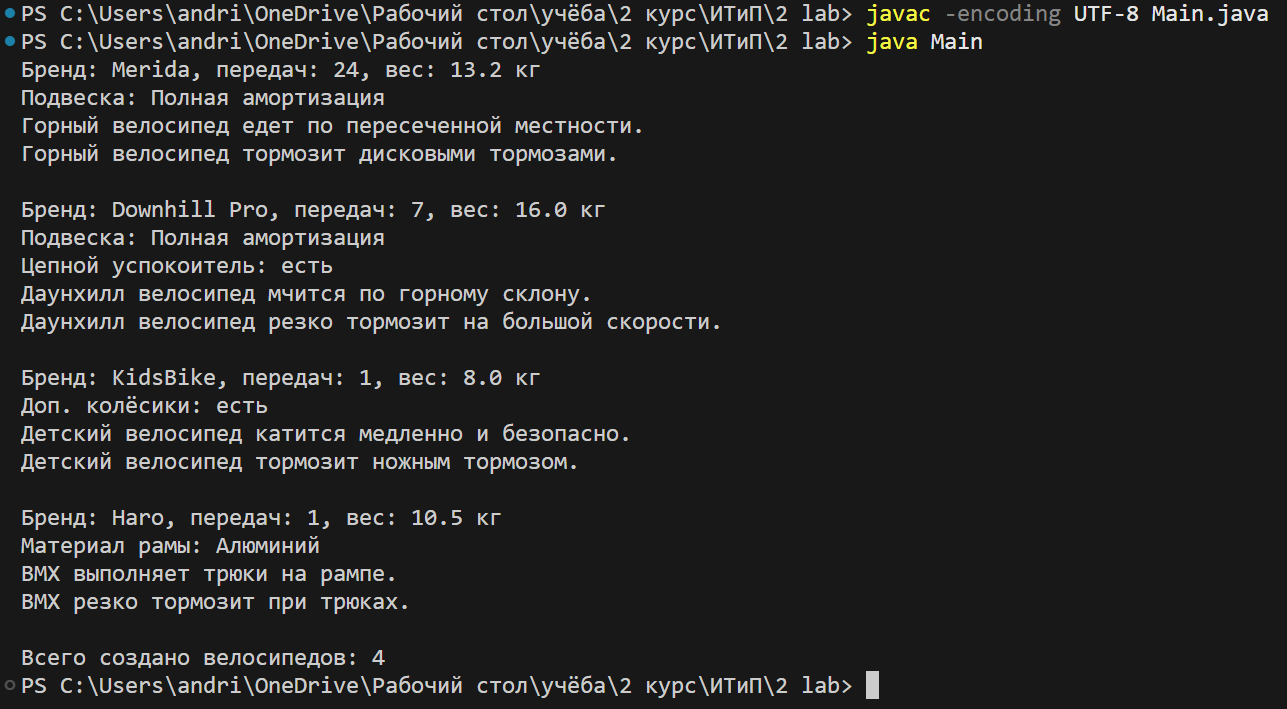
Для демонстрации двухуровневой иерархии создан класс DownhillBike, который наследуется от MountainBike.

* Добавлено поле hasChainGuide.
* Методы ride(), brake() и printInfo() переопределены с учётом особенностей даунхилл-велосипеда.



**IV) Ввод/вывод информации и демонстрация работы.**

В методе main создаются объекты различных велосипедов и демонстрируется их работа. 

Запуск программы: 

**Вывод:** В ходе работы была создана иерархия классов велосипедов с абстрактным базовым классом и несколькими дочерними классами. Были реализованы конструкторы, геттеры и сеттеры, методы для вывода информации и моделирования поведения объектов. Работа позволила закрепить основные принципы ООП: инкапсуляцию, наследование, полиморфизм и абстракцию, а также навыки использования статических переменных для подсчёта объектов.

**Ссылка на GitHub:** <https://github.com/pmr-fiz/Laboratory-2-course/tree/main/Laboratory%202>