

Отчёт

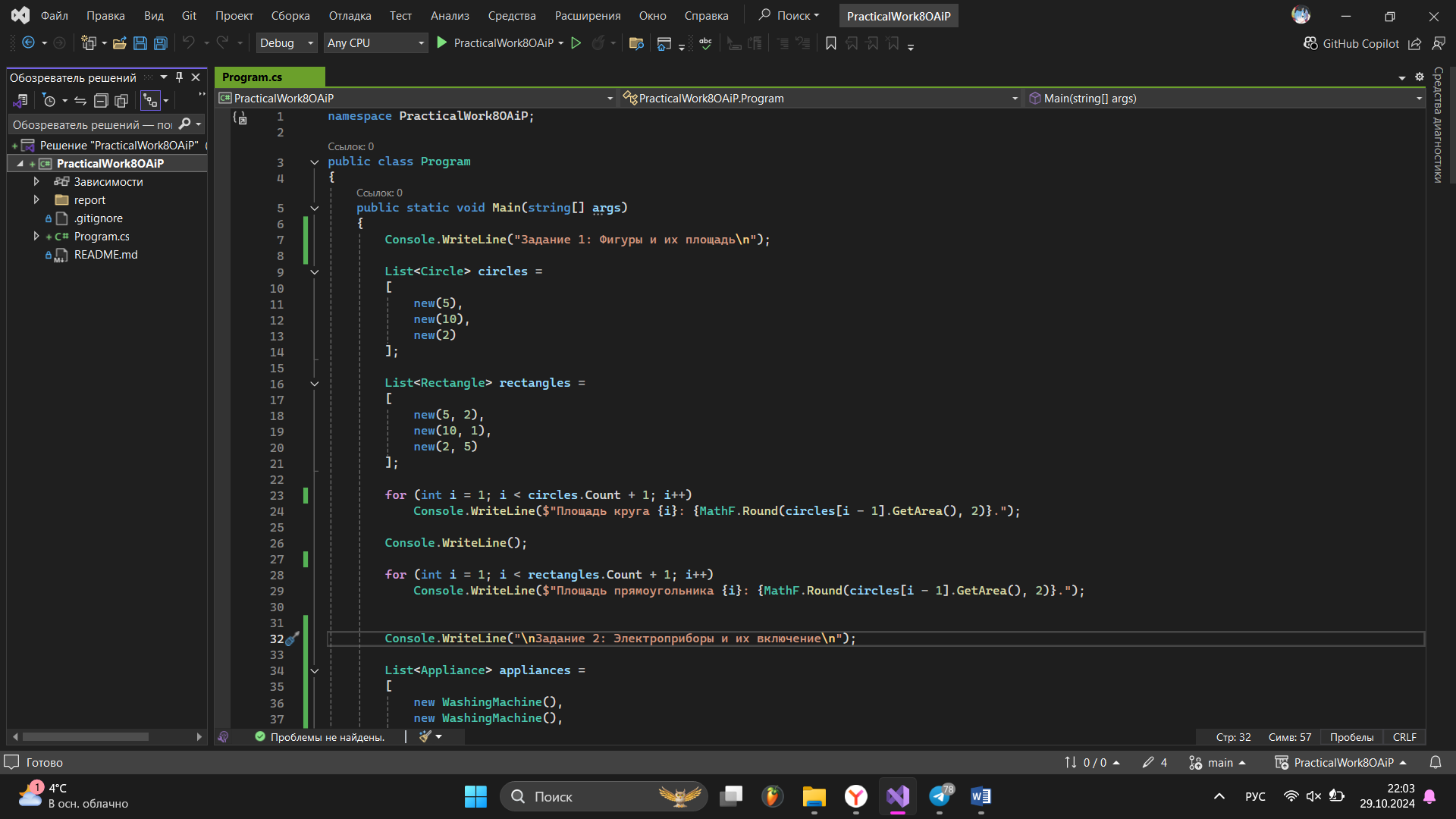
Лабораторная работа №8

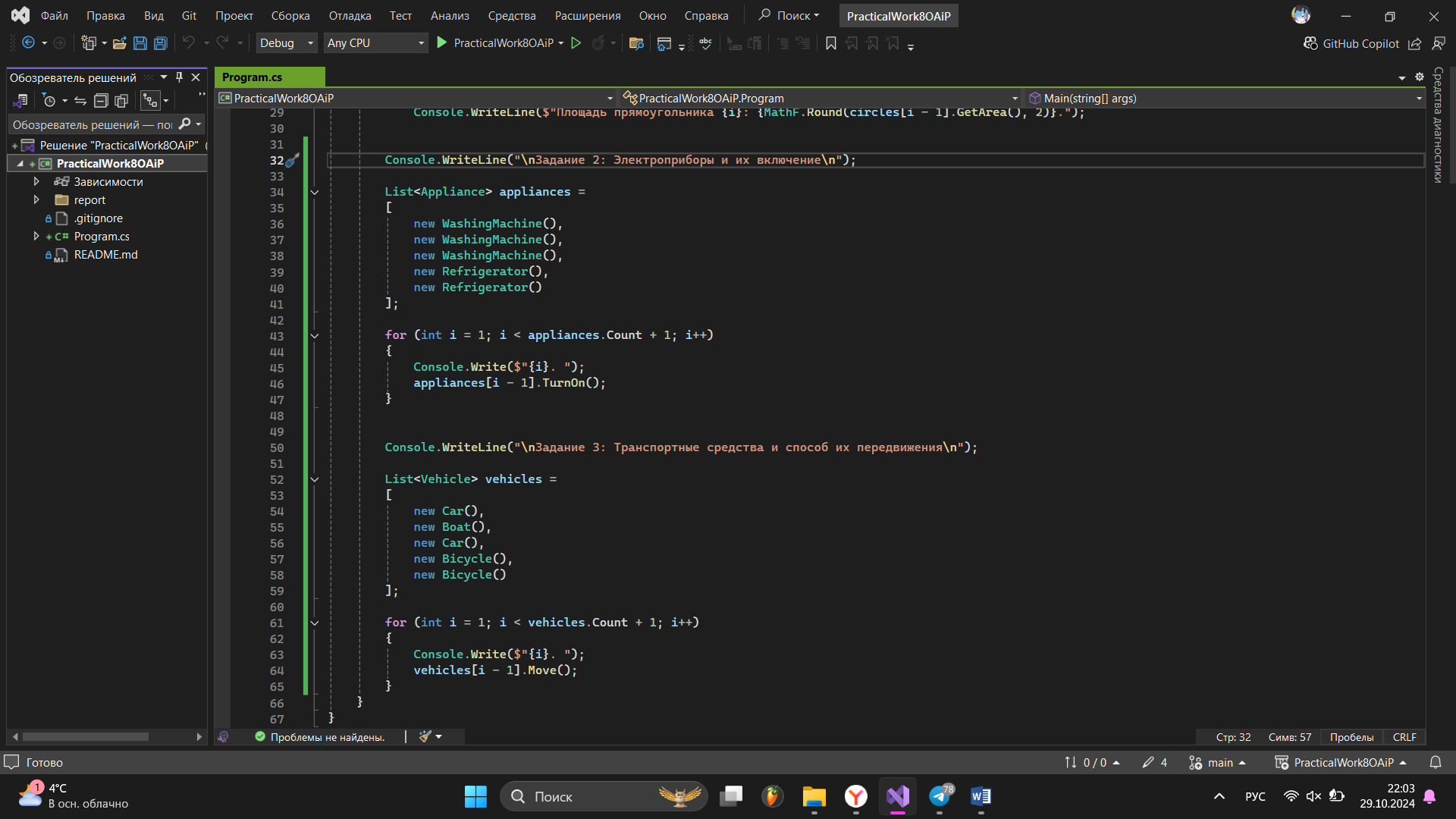
ТЕМА: «Полиморфизм»

«Бедин Владислав Алексеевич»

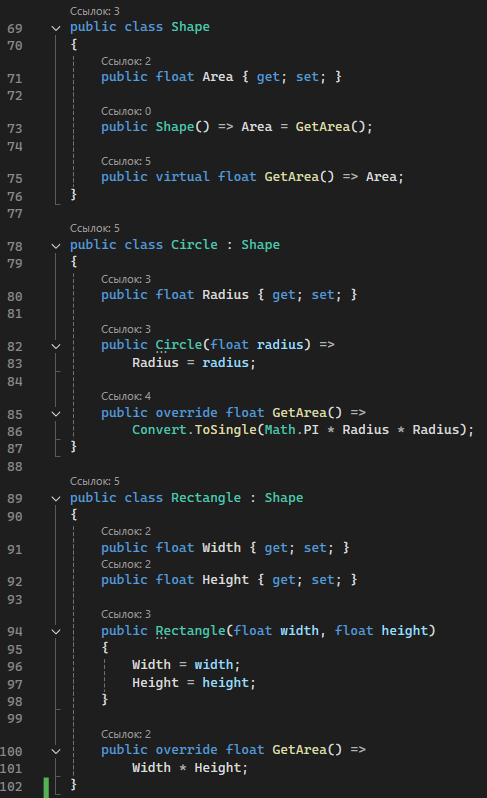
1. Я ознакомился с теоретическими сведениями и изучил механизм полиморфизма в языке программирования C#.
2. Выполнил практические задания «1, 2, 3»
3. СКРИНШОТЫ СДЕЛАННОГО ЗАДАНИЯ

Класс Program:

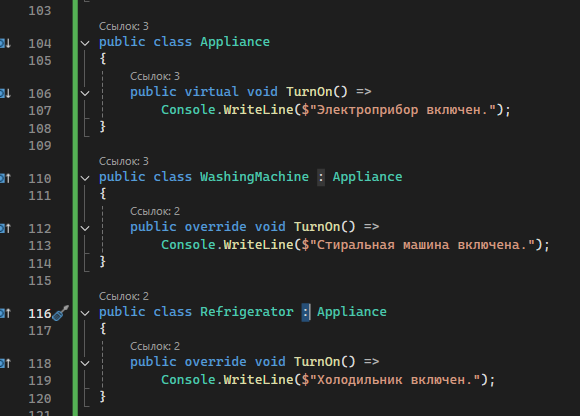




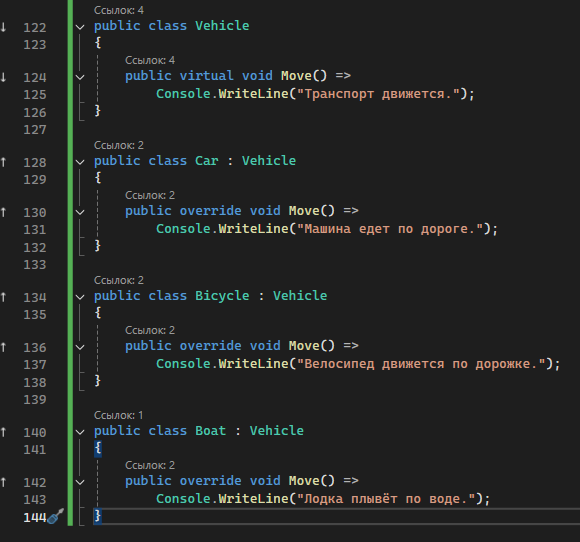
Задание 1: Shape => Circle, Rectangle



Задание 2: Appliance => WashingMachine, Refrigerator



Задание 3: Vehicle => Car, Bicycle, Boat



**Код выполненной работы в формате .txt**



**Ответы на контрольные вопросы**

1. **Наследование** и его **назначение**:

**Наследование** - это механизм, позволяющий создавать новые классы (производные классы) на основе уже существующих (базовые классы), наследуя их свойства и методы.

**Назначение** наследования:

• Повторное использование кода: Избегайте дублирования кода, используя уже существующие классы в качестве основы для новых.

• Иерархическая структура: Создайте иерархию классов, отражающую реальные отношения между объектами.

• Полиморфизм: Обеспечивает возможность работы с объектами разных классов через общий интерфейс.

2. **Виртуальные** методы:

**Виртуальный метод** - это метод, который можно переопределить в производном классе.

Зачем используются виртуальные методы:

• Полиморфизм: Позволяют создавать более гибкие и адаптивные программы, где объекты разных классов могут взаимодействовать через единый интерфейс.

• Расширение функциональности: Производные классы могут изменять поведение виртуальных методов базового класса, добавляя свою специфику.

3. Ключевое слово **override**:

Чтобы переопределить метод из базового класса в производном классе, нужно использовать ключевое слово **override**.

4. Вызов метода при отсутствии переопределения:

Если в производном классе не переопределен виртуальный метод базового класса, будет вызван метод базового класса.

5. **Полиморфизм**:

**Полиморфизм** - это способность объекта принимать разные формы в зависимости от контекста. В C# это достигается за счет использования виртуальных методов и переопределения.

Как помогает работать с объектами разных классов:

• Унификация: Позволяет работать с объектами разных классов через единый интерфейс.

• Гибкость: Код становится более гибким, так как один и тот же код может обрабатывать объекты разных типов.

6. Вызов метода при **полиморфизме**:

При использовании полиморфизма будет вызван метод производного класса, если он переопределен, и метод базового класса, если переопределение отсутствует.

**Вывод**

В ходе выполнения данных заданий я изучил один из основных принципов ООП - **механизм** **полиморфизма**: его синтаксис и применение в языке программирования C#.