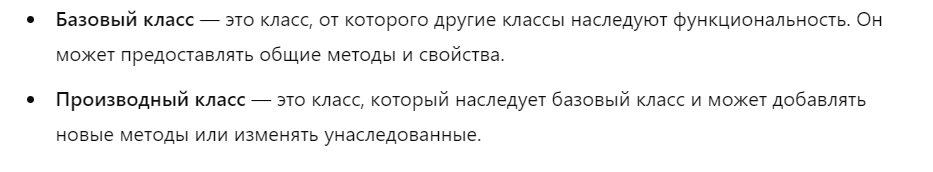
1. Для чего используют статические классы?

Статические классы используют для организации методов и свойств, которые относятся к классу в целом и не зависят от конкретных экземпляров этого класса. Они полезны для хранения вспомогательных методов или данных, которые используются глобально.

2. Что может содержать статический класс?

Статический класс может содержать только статические поля, свойства, методы, и конструкторы. Экземплярные члены и методы в статическом классе запрещены.

3. Что такое производный и базовый классы?



4. Как используют ключевое слово base?

Ключевое слово base используется для обращения к членам базового класса из производного. Это может быть вызов конструктора базового класса или обращение к методам и свойствам базового класса.

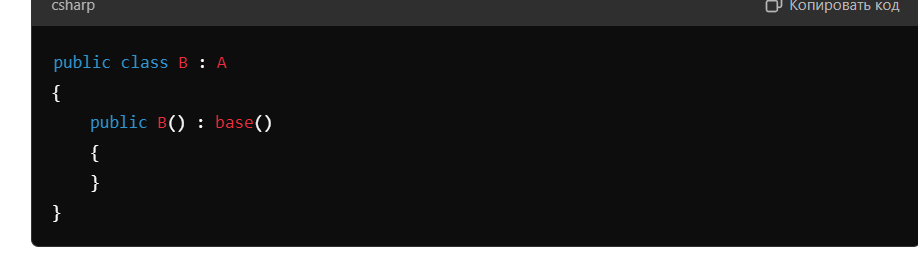
5. В чем заключена основная задача наследования?

Основная задача наследования — это повторное использование кода, уменьшение дублирования и создание гибких архитектур, позволяющих расширять функциональность классов.

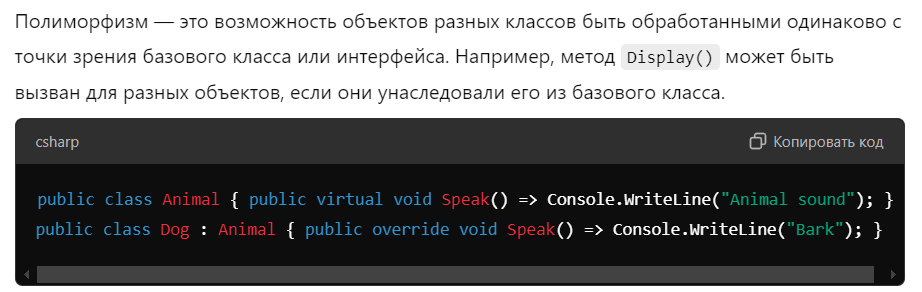
6. Пусть базовый класс содержит метод basefunc(), а производный класс не имеет метода с таким именем. Может ли объект производного класса иметь доступ к методу basefunc()? Если да, то при каких условиях?

Да, объект производного класса может иметь доступ к методу basefunc() базового класса, если этот метод имеет модификатор доступа, позволяющий это (например, public или protected).

7. Напишите объявление конструктора без аргументов для производного класса B, который будет вызывать конструктор без аргументов базового класса A.



8. Что такое полиморфизм? Приведите пример.



9. Определите назначение виртуальных функций.

Виртуальные функции позволяют производным классам переопределять их для изменения поведения базового класса.

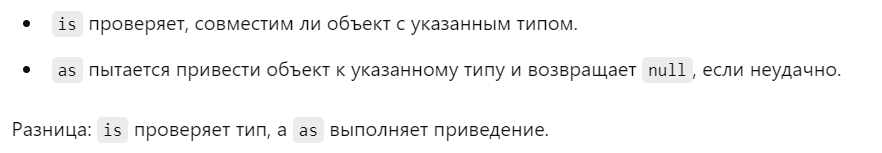
10. Кому доступны переменные с модификатором protected?

Переменные с модификатором protected доступны в самом классе и во всех его производных классах.

11. Наследуются ли переменные с модификатором private?

Нет, переменные с модификатором private не наследуются напрямую, но они доступны через методы и свойства, если они предоставляют доступ к этим данным.

12. As, is – что это, как применяется? В чем между ними отличие?



13. Поддерживает ли C# множественное наследование?

Нет, C# не поддерживает множественное наследование классов, но позволяет наследование от нескольких интерфейсов.

14. Можно ли запретить наследование от класса?

Да, для этого используют модификатор sealed.

15. Можно ли разрешить наследование класса, но запретить перекрытие метода?

Да, для этого метод объявляют с модификатором sealed в производном классе.

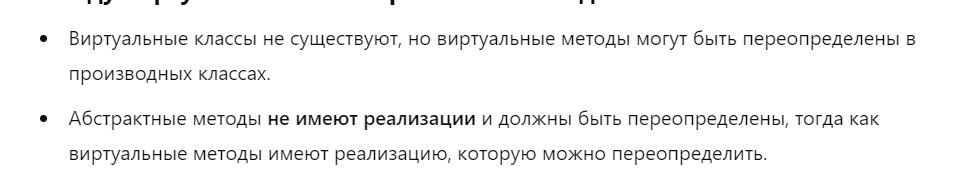
16. Что такое абстрактный класс?

Абстрактный класс — это класс, который не может быть создан как экземпляр и может содержать абстрактные методы без реализации, которые должны быть реализованы в производных классах.

17. В каком случае вы обязаны объявить класс абстрактным?

если класс имеет хотя бы одно абстрактное свойство или метод, то он должен быть определен как абстрактный

18. В чем разница между абстрактными и виртуальными классами? Между виртуальными и абстрактными методами?



19. Какие компоненты класса могут быть виртуальными?

Виртуальными могут быть методы, свойства, индексаторы.

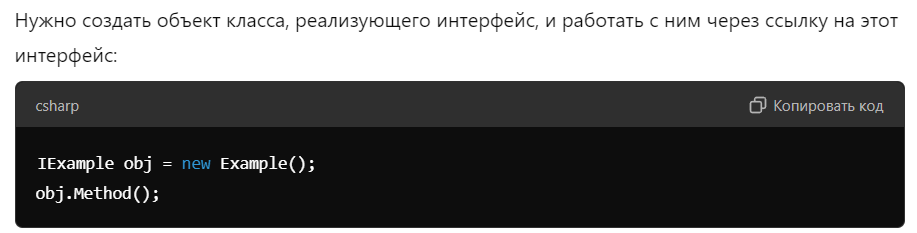
20. Что такое интерфейс?

Интерфейс - перечень сигнатур методов, предназначенных для реализации.

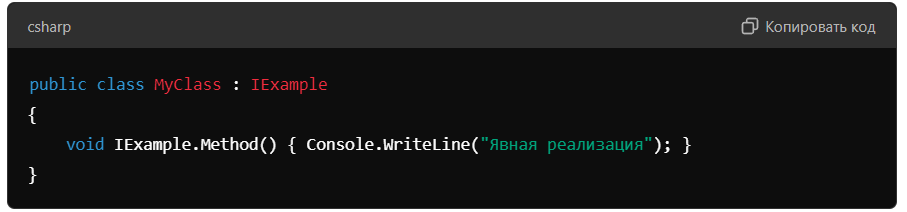
21. Что может содержать интерфейс?

Интерфейс может содержать объявления методов, свойств, событий и индексаторов, но не может содержать их реализацию (до C# 8.0).

22. Как работать с объектом через унаследованный интерфейс?



23. Приведите пример явной реализации интерфейса.



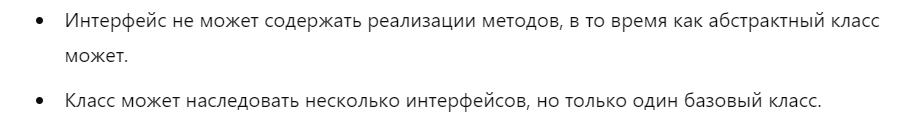
24. Почему нельзя указать модификатор видимости для методов интерфейса?

Все методы интерфейса неявно являются public, так как интерфейс описывает контракт, который должен быть доступен для реализации.

25. Можно ли наследовать от нескольких интерфейсов?

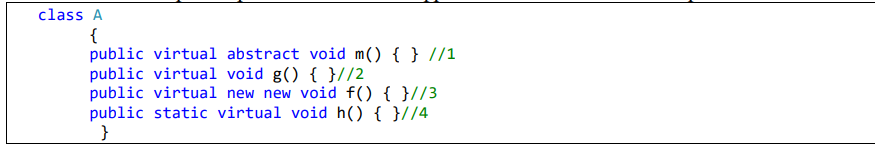
Да, в C# можно наследовать от нескольких интерфейсов.

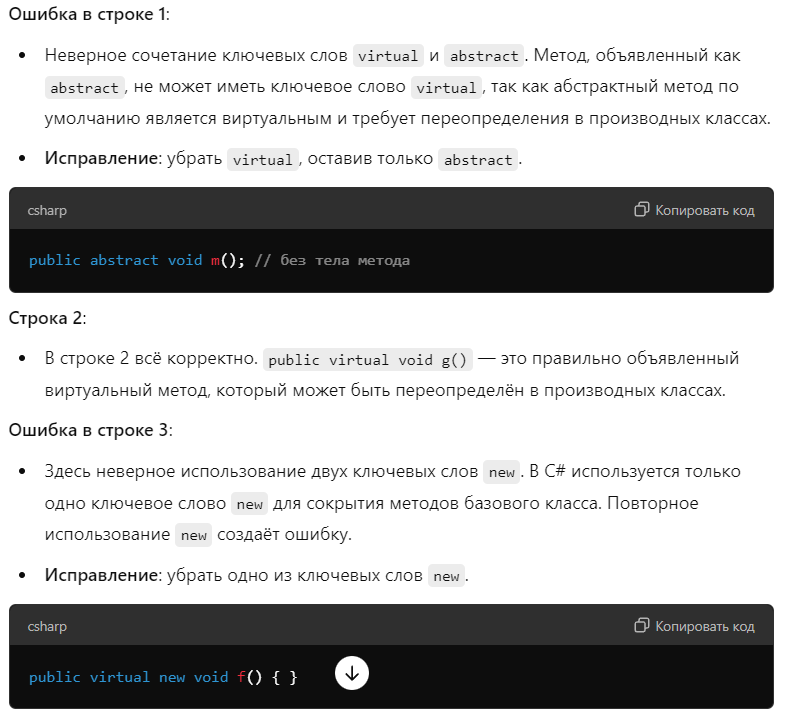
26. Назовите отличия между интерфейсом и абстрактным классом.

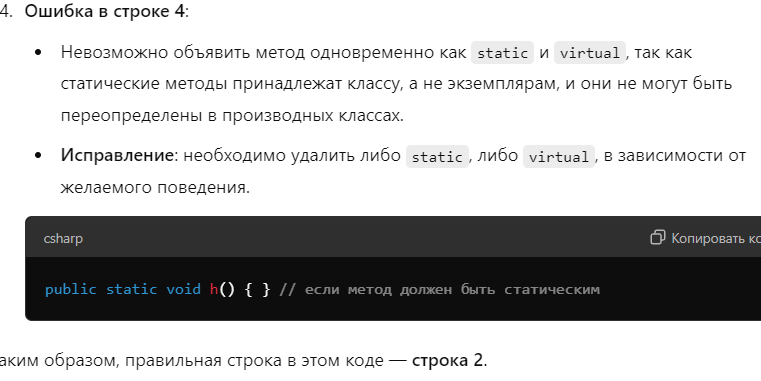


27. Для чего используются стандартные интерфейсы ICloneable, IComparable, IComparer, lEnumerable?

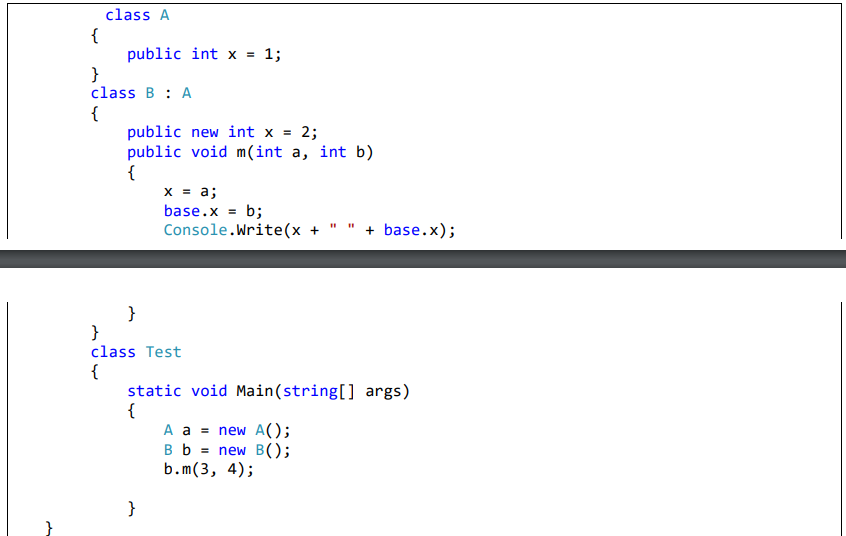
Эти интерфейсы используются для реализации стандартного поведения, такого как клонирование объектов (ICloneable), сравнение объектов (IComparable и IComparer), и перебор коллекций (IEnumerable).

28. В какой строке приведенного ниже фрагмента листинга не содержится ошибки? 



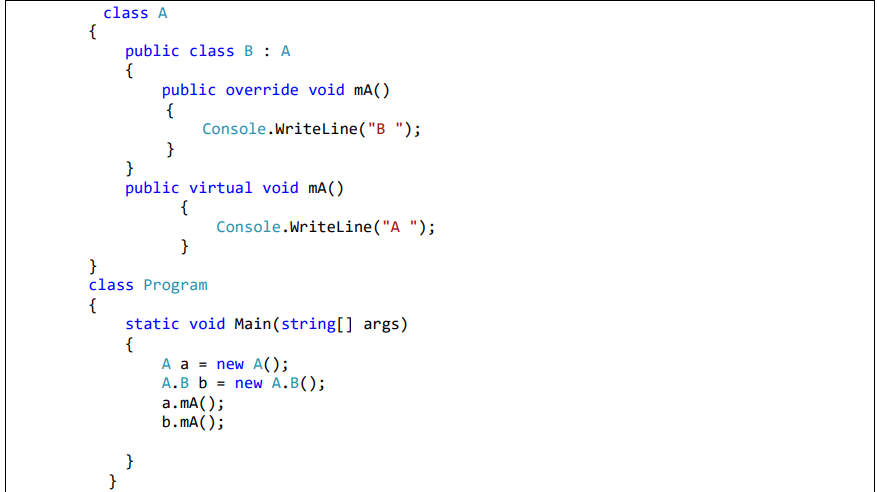


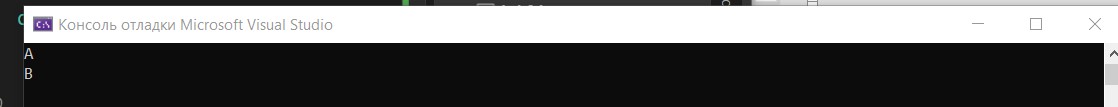
29. Что будет выведено на консоль в результате выполнения следующего фрагмента?

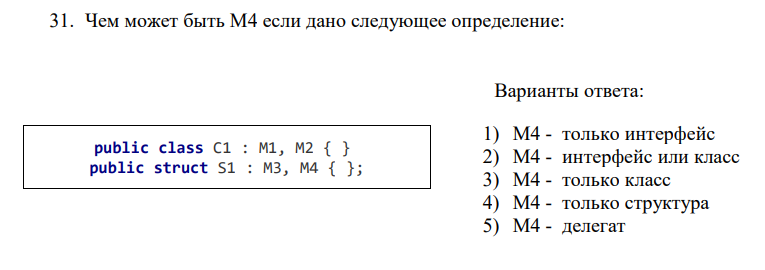


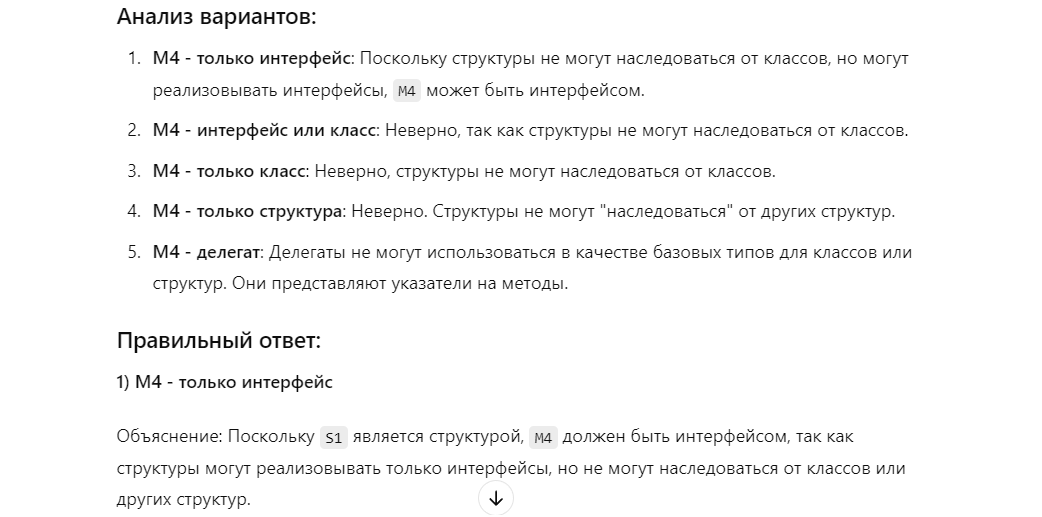
3 4

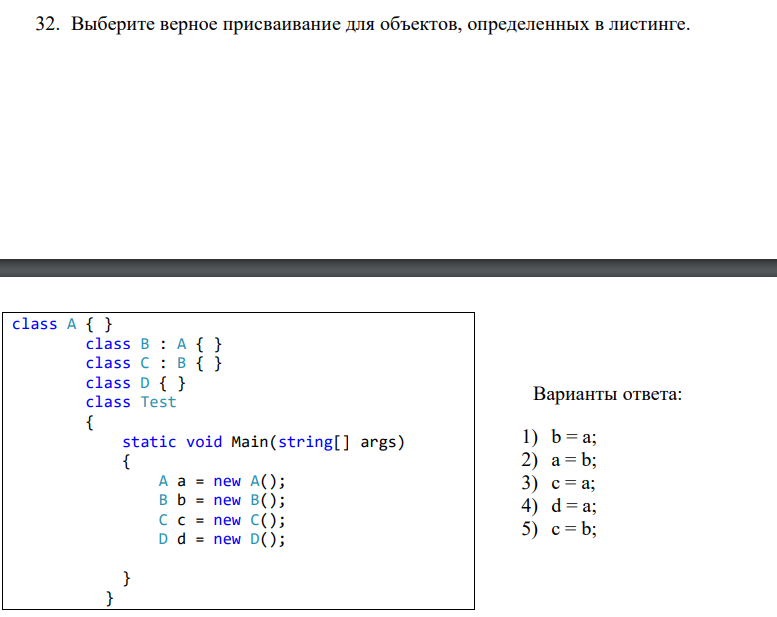
30. Что будет выведено на консоль в результате выполнения следующего фрагмента кода?

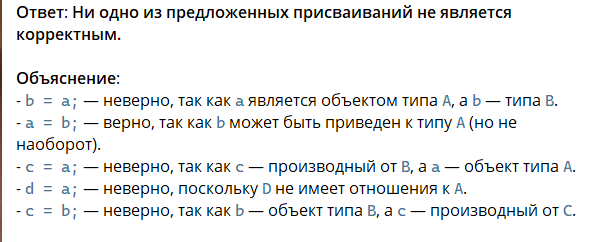


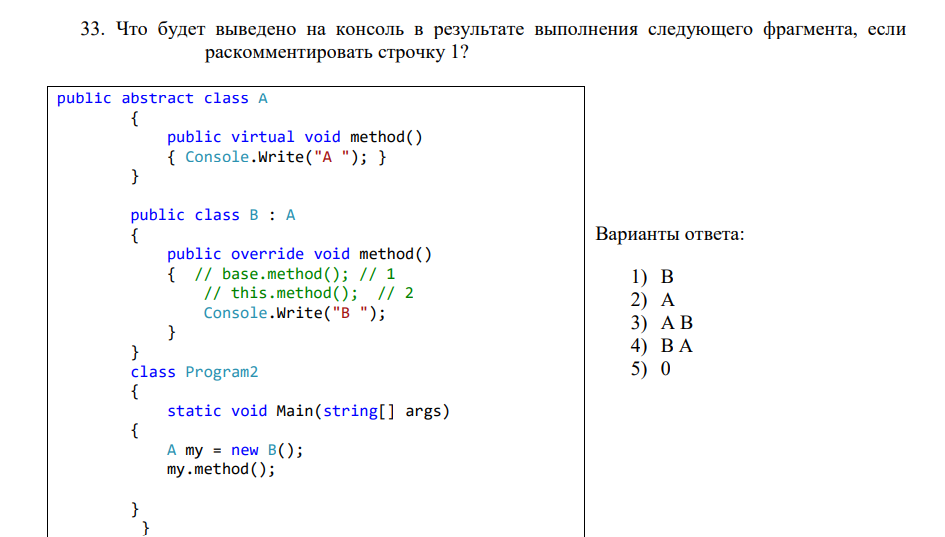
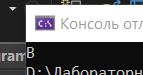


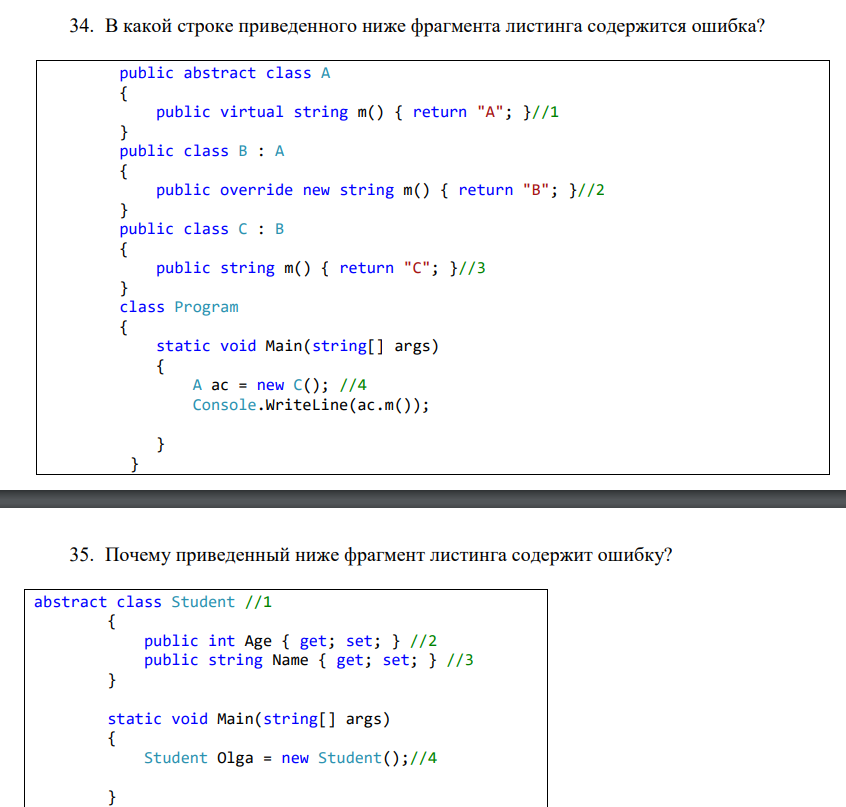


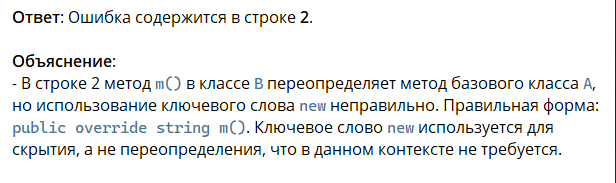


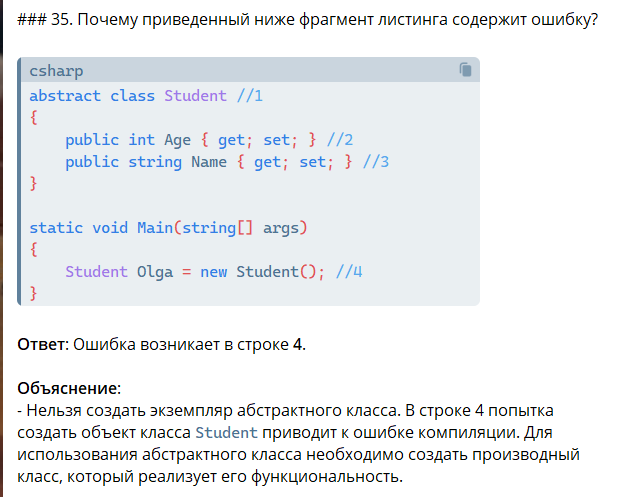


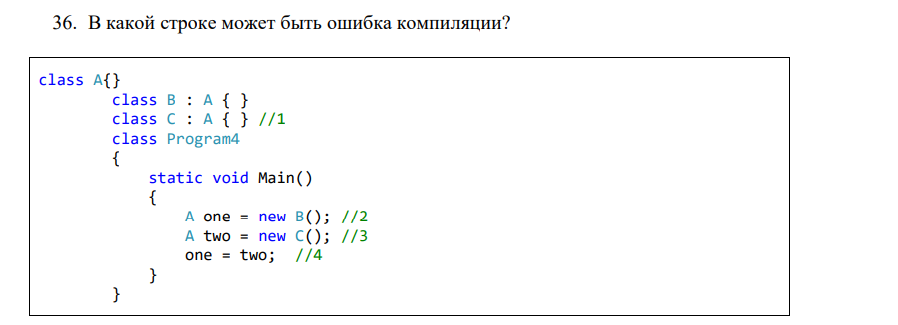


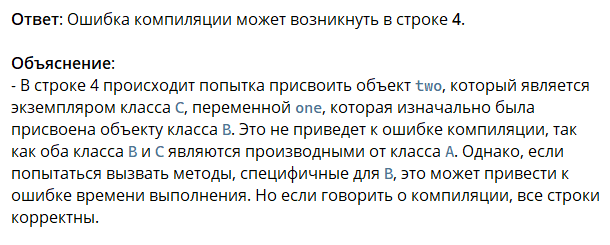


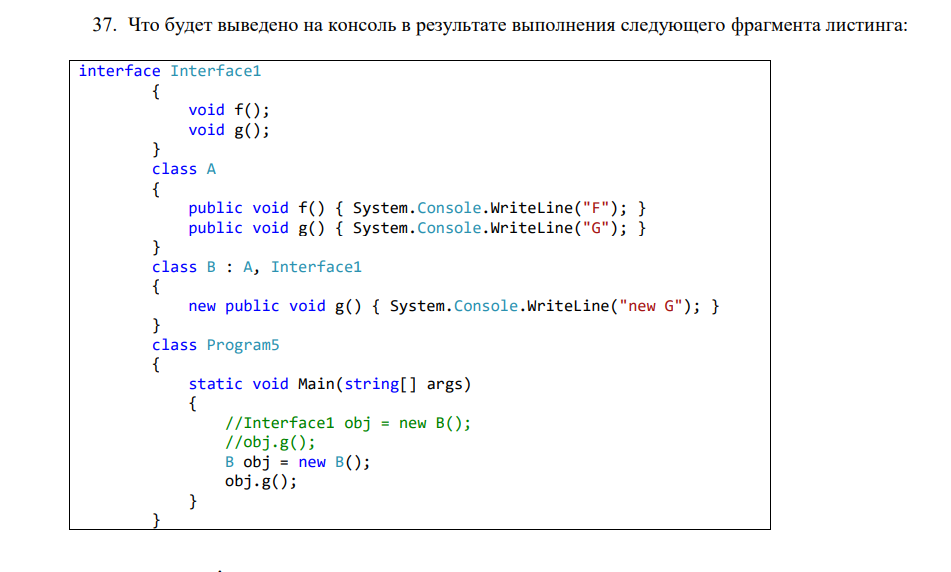












New G