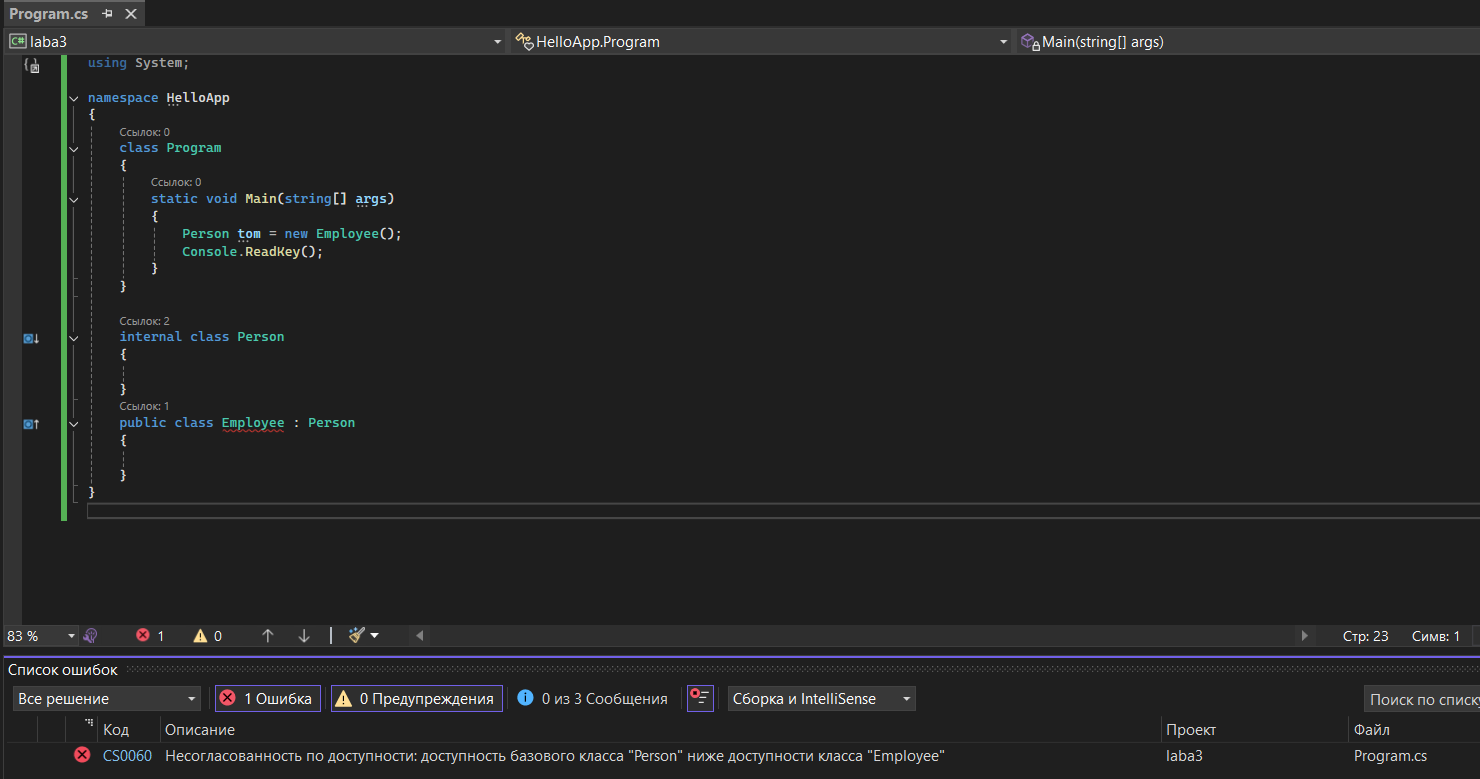
**Практическая работа №3 « Задачи с созданием классов».**

**Выполнил студент группы Юнязов Богдан Денисович.**

**1 Задание:**

****

**Ответ:** Программа не компилируется из-за несоответствия модификаторов доступа. Класс Person объявлен как internal, а класс Employee как public и наследует от Person. Это недопустимо, так как базовый класс (Person) имеет более строгий модификатор доступа (internal), чем производный класс (public). Производный класс не может быть более доступным, чем его базовый класс.

**2 Задание:**

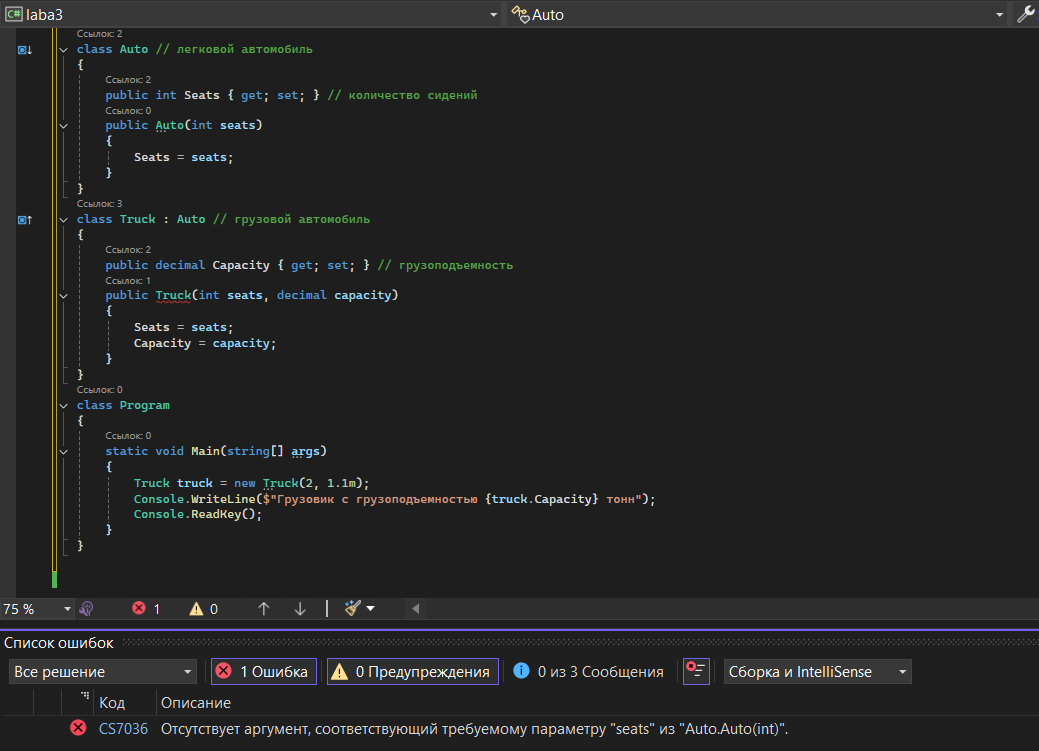
Будут выполнены следующие конструкторы в таком порядке:

1. public Person(string name) : this(name, 18) - вызывает следующий конструктор
2. public Person(string name, int age) - базовый конструктор с параметрами
3. public Employee(string name, string company) : base(name) - конструктор Employee

**3 Задание:**

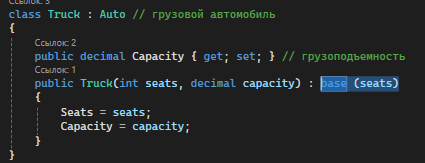
Чтобы запретить наследование от класса, нужно пометить его ключевым словом **sealed.**

**4 Задание:**

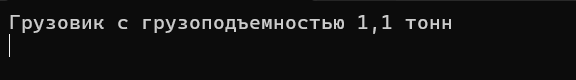
****

в классе Truck вы не вызывается конструктор базового класса Auto, что может привести к неправильной инициализации свойств.

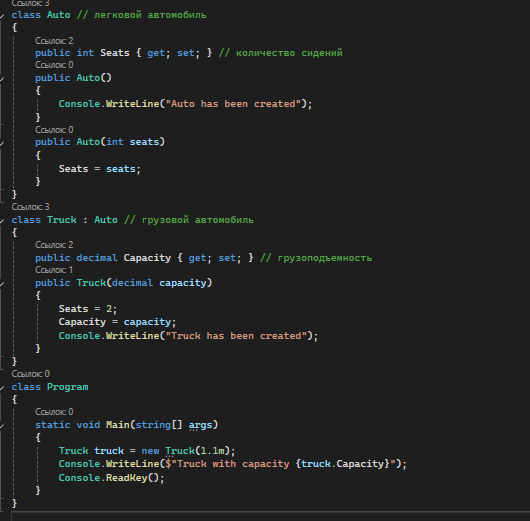
**Чтобы исправить это, вам нужно вызвать конструктор базового класса с помощью ключевого слова base.**

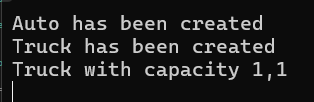
****

Результат:

****

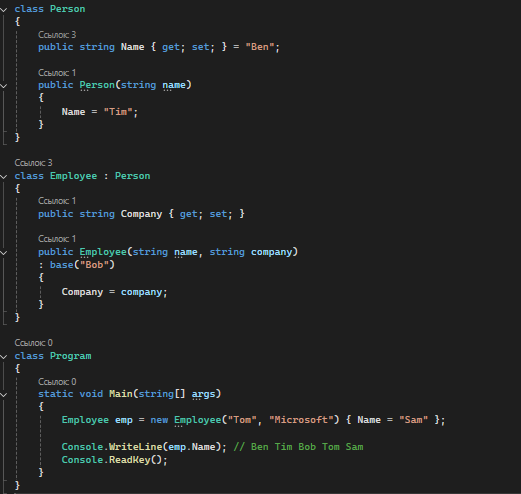
**Задание 5:**

****

****

Сначала вызывается конструктор по умолчанию базового класса Auto(), затем выполняется конструктор Truck.

**Задание 6:**

****

****

Выведет **"Sam".**

Порядок инициализации:

Инициализатор свойства Name = "Ben" (но он будет перезаписан)

Вызов конструктора базового класса base("Bob")

В конструкторе Person устанавливается Name = "Tim"

В конструкторе Employee устанавливается Company

После создания объекта применяется инициализатор { Name = "Sam" }, который перезаписывает предыдущее значение

Таким образом, последнее присваивание (Name = "Sam") перезаписывает все предыдущие значения.