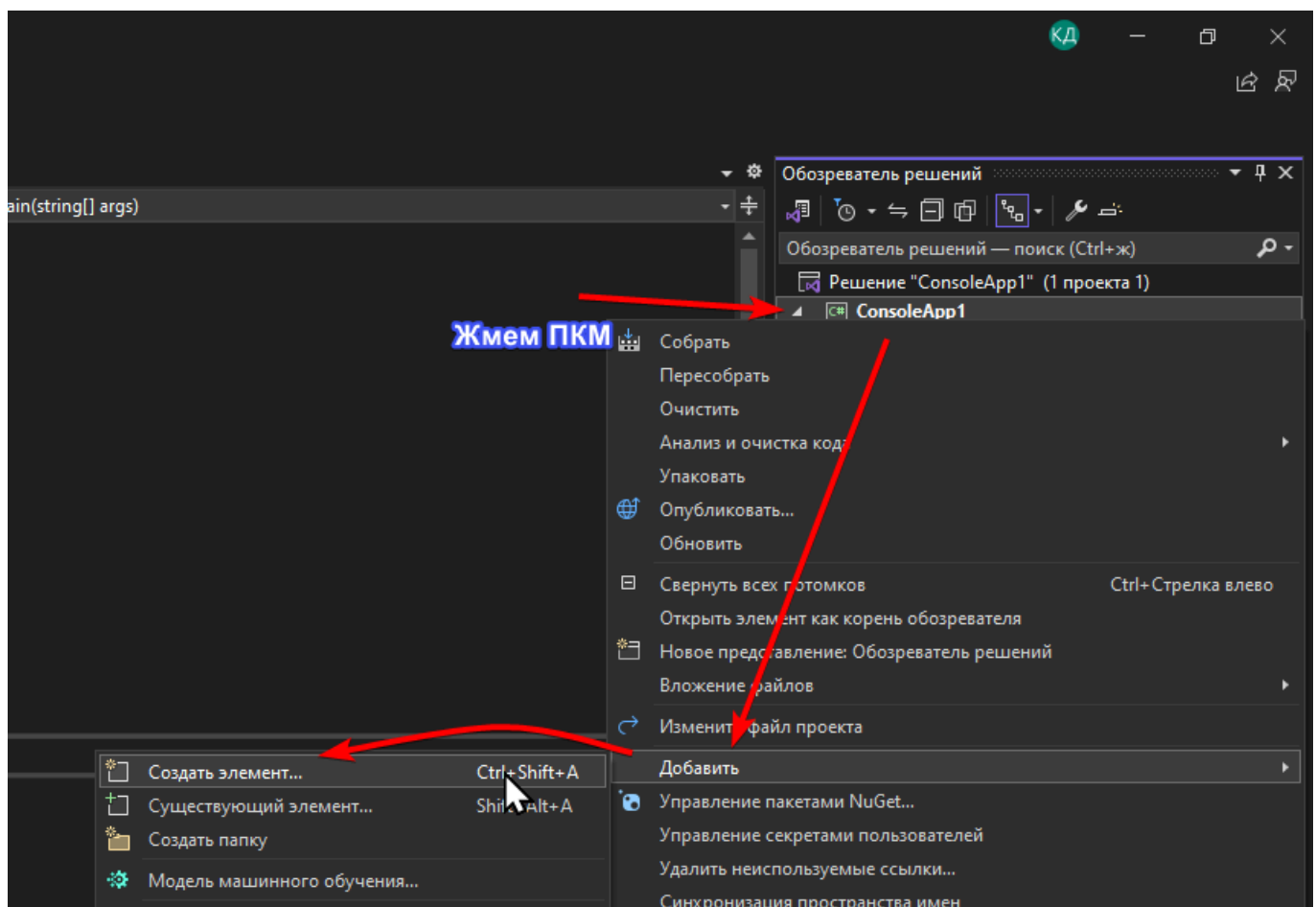




## Практическая работа 8.

### Классы и Объекты

В данной практической работе необходимо создавать Классы. Классы создаются в отдельном файле формата .cs



В открывшемся окошке пишем название нового класса, **.cs обязательно оставляем.**



Добавить новый элемент

ConsoleApp1/

Person.cs

Показать все шаблоны

Добавить

Отмена

---

## Задание 1

Объявите класс **Person**, который описывает человека.

Определите **поля** класса:

- имя - **name (string)**,
- возраст - **age (int)**,
- работа - **job (string)**.

Объявите **методы** класса Person:

- **Meet**, который пишет сообщение в консоль “Привет! Меня зовут {имя}! Мой возраст - {возраст}.”.
- **GoToJob**, который пишет сообщение “{Имя} пошел на работу. Его работа - {работа}.”

**Значение полей класса необходимо передавать в конструктор.**

Создайте три объекта класса Person внутри метода Main. У каждого вызовите методы Meet и GoToJob.



---

## Задание 2

Объявите класс **Bank**.

Объявите поля класса Bank:

- текущий баланс счета - **balance (decimal)**
- владелец счета - **owner (Person)**.

Объявите метода класса Bank:

- **AddMoney(decimal)** - увеличивает баланс счета, прибавив значение переданное в параметр.
- **Remove(decimal)** - уменьшает баланс счета, вычтя значение переданное в параметр.
- **SendMoney(decimal, Bank)** - уменьшает свой баланс и увеличивает баланс счета, который передан в параметр метода, на значение, которое передан в параметр метода. Нельзя выполнять перевод, если владелец счета отсутствует (равен null).

**Начальный баланс задается через параметр в конструкторе.**

Создайте в Main два счета и два человека. В main привяжите к каждому счету человека. Увеличьте каждый счет на 10000. Выполните передачу с одного счета на другой.



Добавьте проверки, чтобы баланс счета не опускался ниже 0. В случае нехватки денег операция перевода должна отменяться и необходимо выводить сообщение об ошибке.

Выводите сообщения о пополнении, снятии и кто кому перевел деньги.

“Пополнения счета +{значение счета}!”

“Снятие со счета -{значение счета}!”

“{имя получателя} получил перевод +{сумма пополнения} от {имя отправителя}!”

---

### Задание 3

Объявите класс **Car**.

Объявите поле класса Car:

- скорость - **speed (int)**
- ускорение - **acceleration (int)**.

Объявите метод класса Car:

- **Accelerate**, который будет увеличивать скорость на значение ускорения.
- **Break**, который будет уменьшать скорость на значение удвоенного ускорения.

**Скорость не должна опускаться ниже 0 и подниматься выше 200.**

Объявите класс **Radar**.

Объявите поле класса Radar: **speedLimit** (ограничение скорости).



Объявите метод класса **Radar**:

**Check**, который принимает в качестве параметра машину и проверяет её текущую скорость. Если скорость превышает заданное в радаре ограничение, то выводится сообщение о нарушении.

В методе Main создайте объекты Car и Radar. В цикле на 20 итераций выполните следующие инструкции:

1. Машина случайным образом выполняет одну из двух действий - набирает скорость или тормозит.
2. Радар проводит проверку машины на скорость.