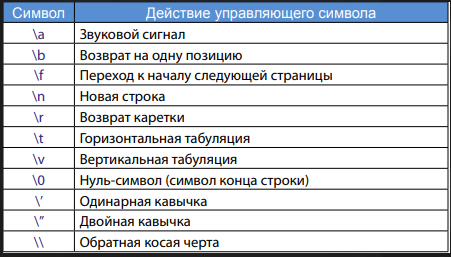
Литералы ?

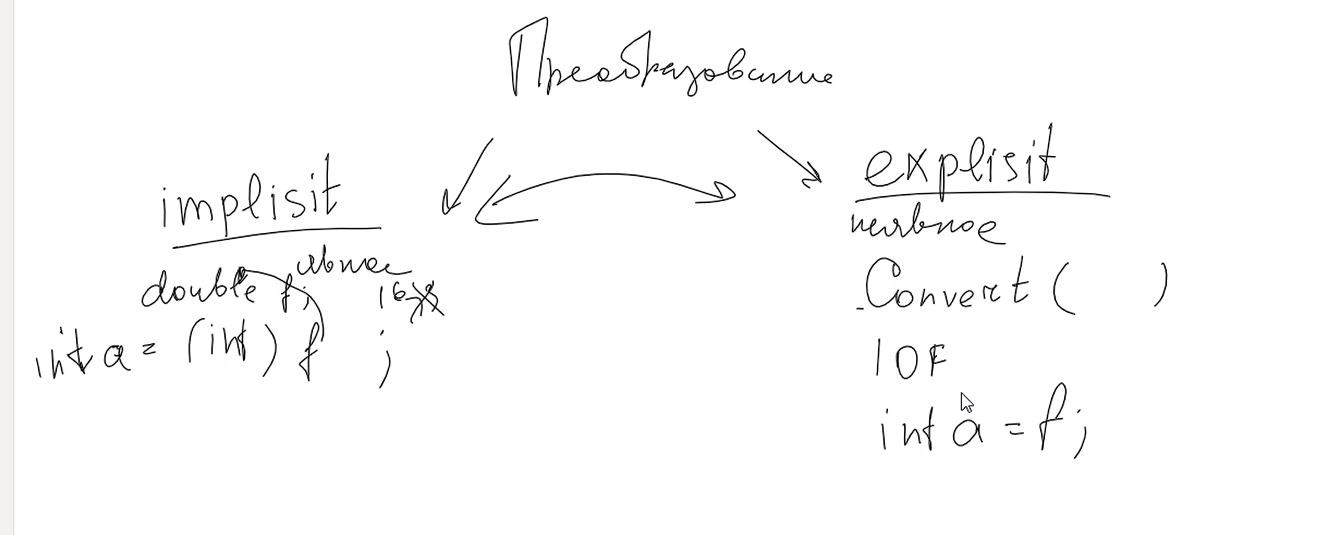
Явные и неявные преобразования

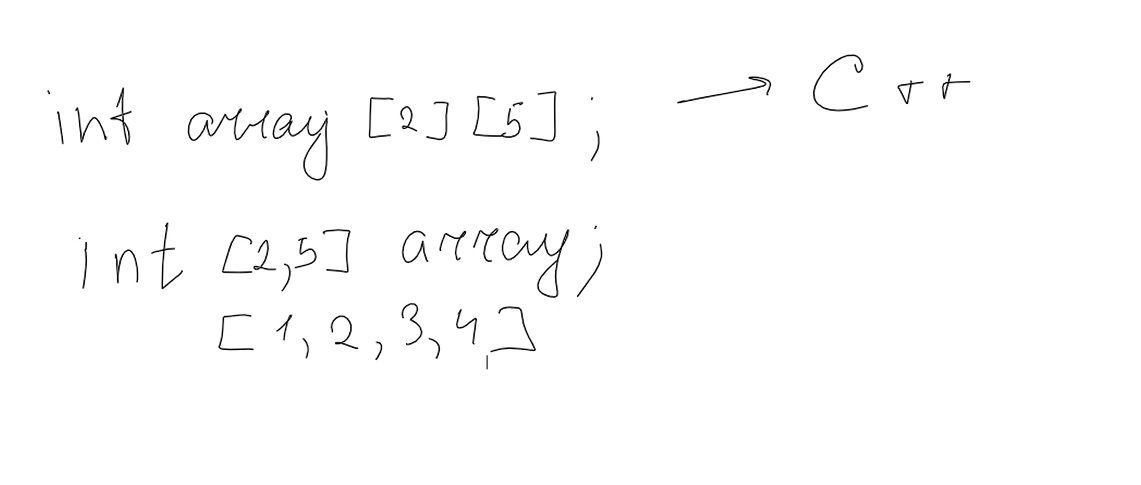
Esacpae-Последовательности

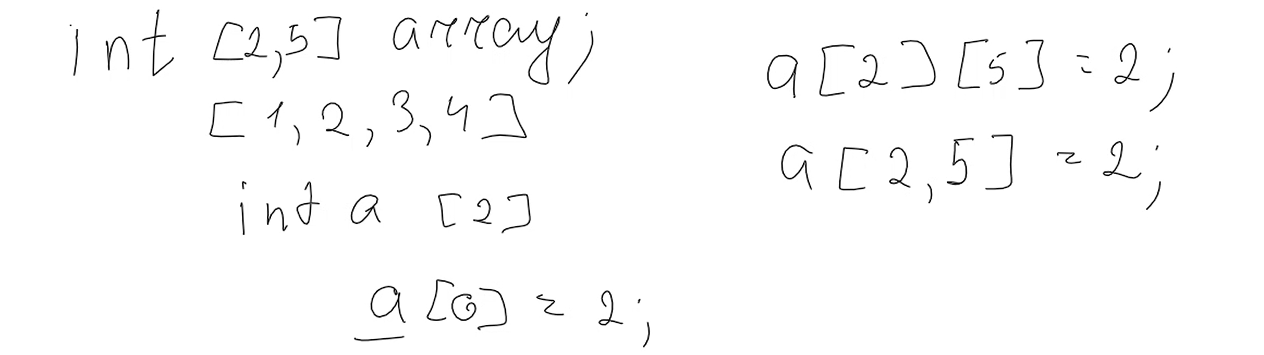


Строгоя типизация

<https://habr.com/ru/articles/172091/>



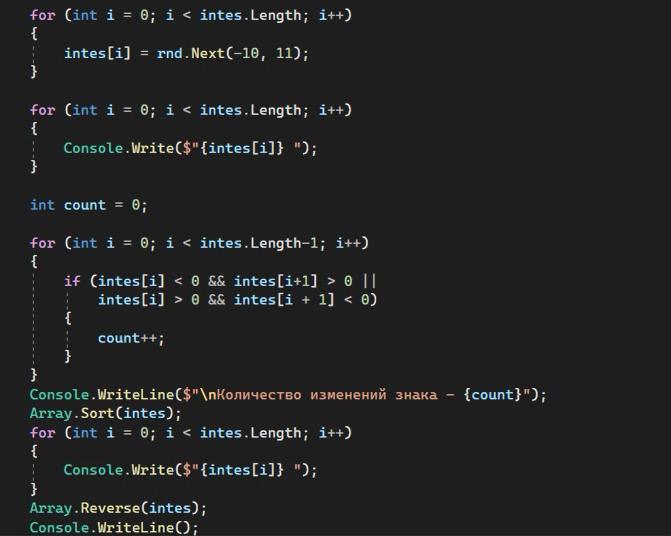


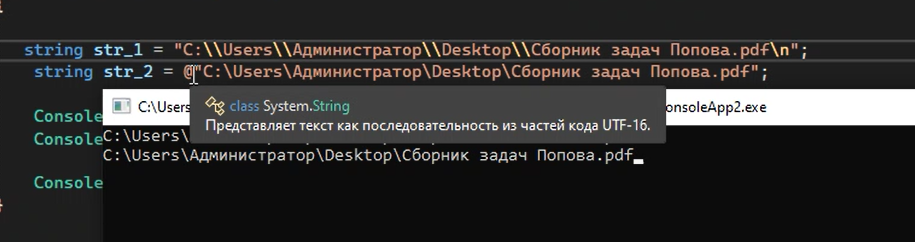


Снипет

<https://habr.com/ru/articles/724444/>

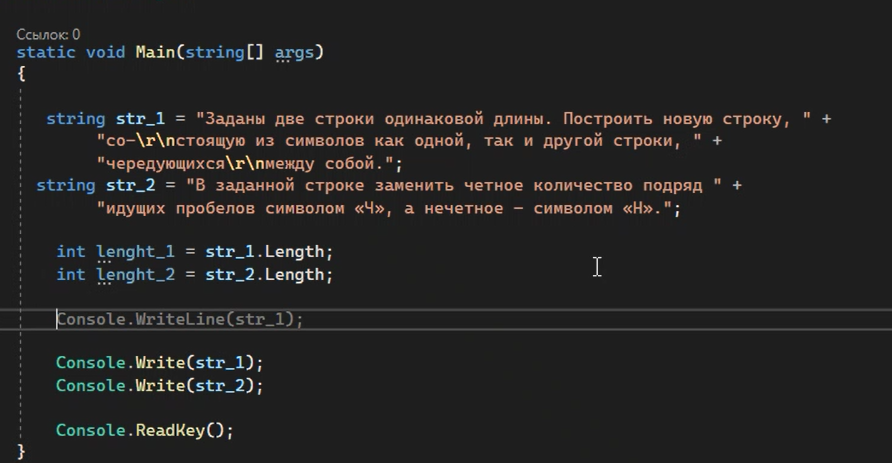


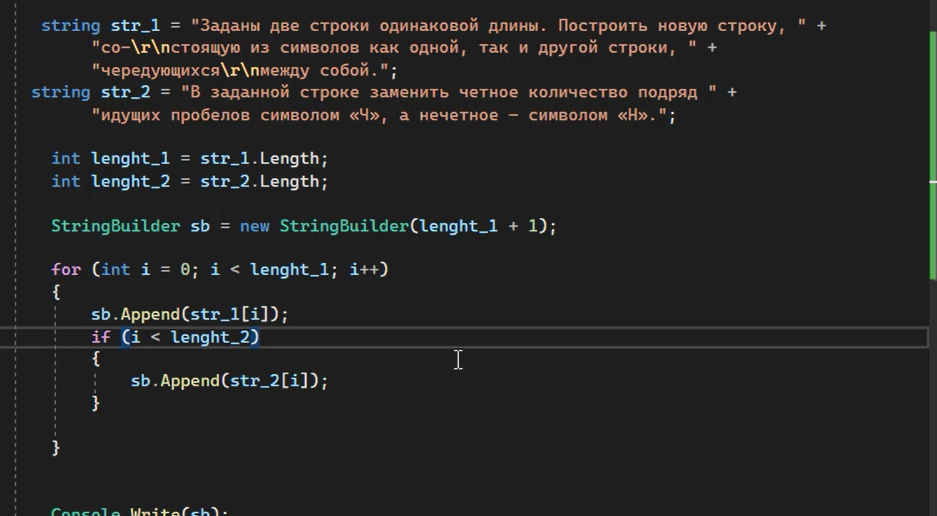




Сериализация

Стрингбилдер



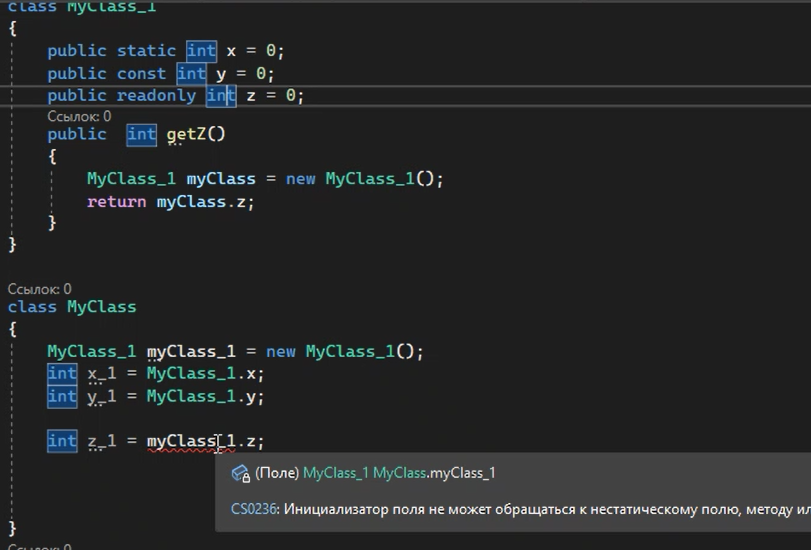


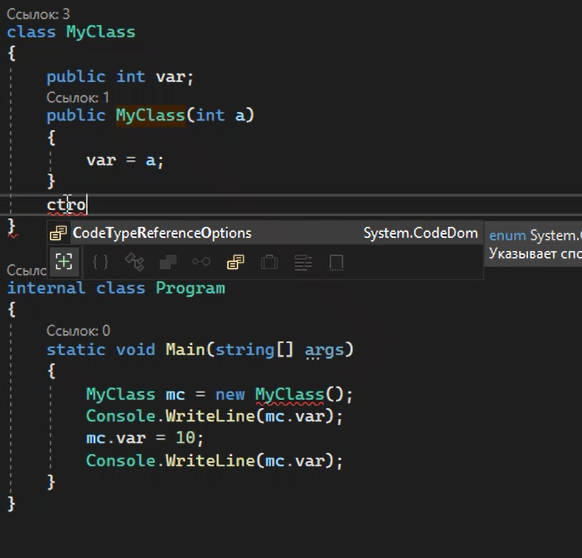
Перечисление

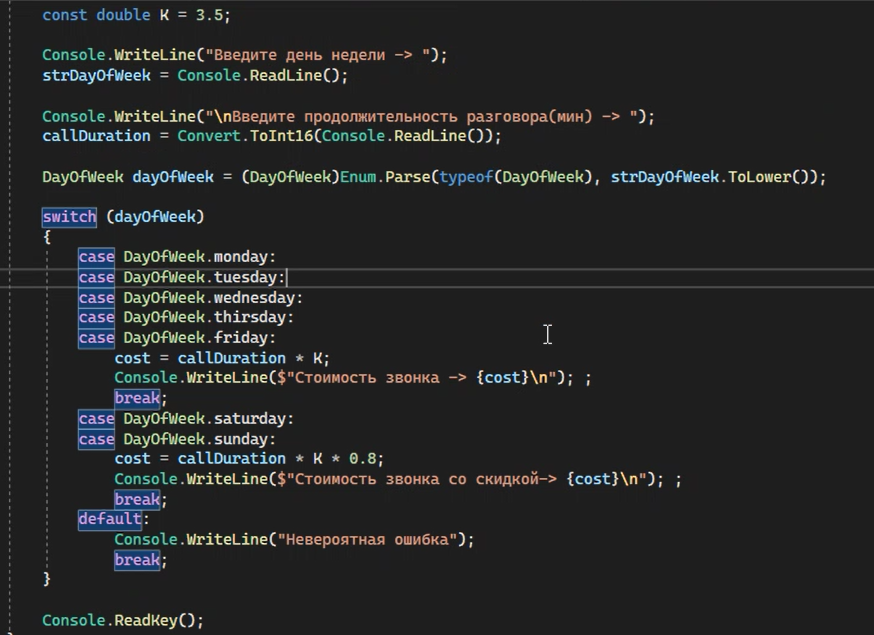
Структура

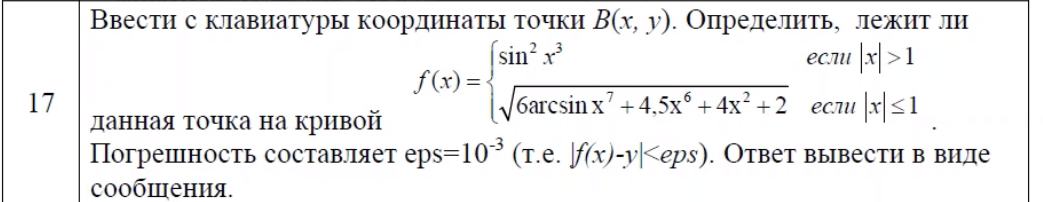
Internal

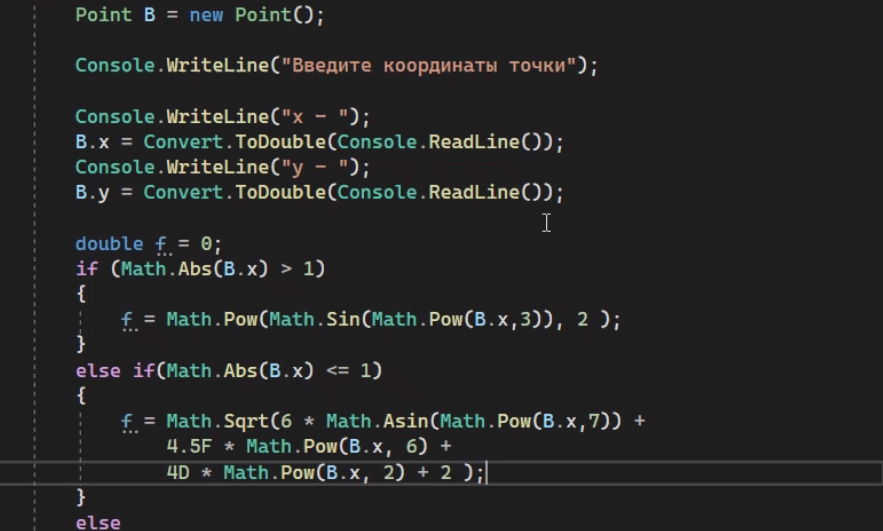
Какие бывают поля?

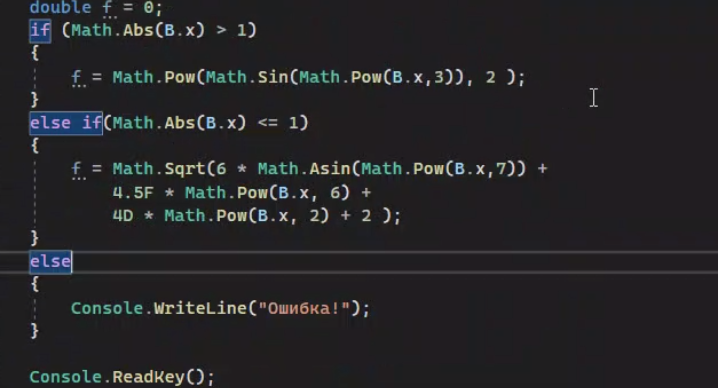


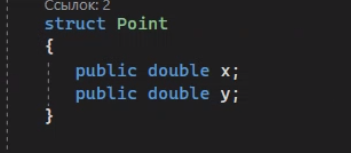


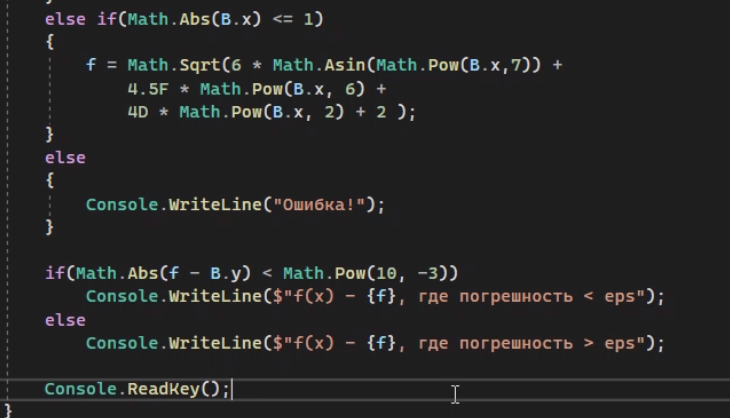


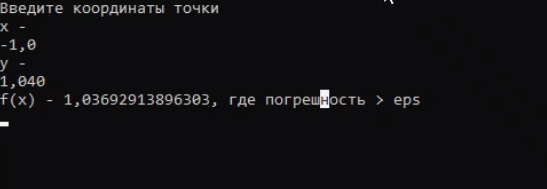












**Что такое пространство имен?**

Пространство имен можно рассматривать как "семейство" типов. Класс содержит код, реализующий тип. Сами классы организуются в пространства имен, чтобы избежать конфликтов имен. Это связано с тем, что при наличии тысяч классов существует вероятность того, что какие-либо из них будут иметь одинаковые имена. Применение пространства имен позволяет гарантировать, что каждый класс будет иметь уникальное *полное* имя.

<https://github.com/AlexUstosu/c_sharp_task>

## Поиск нужных компонентов в библиотеке классов .NET

Как найти нужные для вашего приложения классы и методы среди множества доступных в библиотеке?

Во-первых, при разработке ПО на языке C# вы, скорее всего, будете использовать ограниченный набор классов. В зависимости от типа проектов, с которыми вы работаете, вы будете чаще использовать отдельные разделы библиотеки классов .NET и реже заглядывать в другие. Никто, даже сотрудники корпорации Майкрософт, не может знать абсолютно все.

Во-вторых, по мере необходимости вы, скорее всего, будете использовать свою любимую поисковую систему, чтобы найти записи блога, статьи или форумы, где другие пользователи должны делать то, что вы хотите сделать. Некоторые сторонние организации могут предоставлять рекомендации и даже примеры кода, которые можно попробовать в работе.

Также вы, скорее всего, будете изучать документацию Майкрософт по различным методам, чтобы понять, каким образом они работают, какие ограничения имеют, при каких условиях возвращают исключения или ошибки, а также как избежать потенциальных проблем при их использовании. В нашей документации вы найдете всю необходимую информацию о библиотеке классов .NET. Наша команда авторов документации тесно сотрудничает с разработчиками ПО для библиотеки классов .NET, чтобы гарантировать ее максимальную точность и достоверность.

Наконец, вы начнете экспериментировать с небольшими проектами, чтобы узнать больше о применении и принципах работы классов и методов.

Сталкиваясь с незнакомыми задачами, все разработчики ПО следуют примерно одному и тому же процессу. Несмотря на неизбежные трудности, это очень увлекательное и познавательное занятие.

* Библиотека классов .NET содержит обширный набор функциональных возможностей, которые можно использовать посредством простых ссылок на нужные классы и методы.
* В состав библиотеки классов .NET входят и типы данных. В C# используются лишь псевдонимы этих типов.
* Применение *пространств имен* позволяет исключить конфликты имен между классами из библиотеки классов .NET.
* ля начала введите имя Consoleкласса .
* Добавьте оператор доступа к члену . , символ .
* Добавьте имя WriteLineметода , .