TUPLES (КОРТЕЖИ) В TS

Для разных задач в коде могут использоваться разные структуры данных! Это зависит от ситуации, определенных особенностей проекта или личных предпочтений разработчика

Кортежи - это структура данных, которая существует **только** в TS. Она необходима для записи набора данных в строго определенном порядке:

```
1   const userDataTuple: [boolean, number, string] = [true, 40, "John"];
```

*На данный момент есть предложение добавить tuples в стандартный JS, но оно пока на рассмотрении

Синтаксисом они похожи на массивы. Именно в них и преобразуются после компиляции. Но могут содержать разные типы данных, которые записываются в аннотацию **в строгом порядке**

Основной минус - мы не сразу можем понять что это за данные и к чему они относятся. Нужно иметь описание, документацию или просто помнить почему именно такой порядок данных. За счет этого данная структура может встречаться нечасто

РАБОТА С TUPLES В TS

Так как кортеж технически - это массив, то на нем работают все методы и свойства массивов. Но есть ограничения

```ts
const userDataTuple: [boolean, number, string] = [true, 40, "John"];
userDataTuple.push(50);
userDataTuple[3];
```

Метод сработает, но в TS коде вы не сможете обратиться к 4му элементу кортежа. Его нет в аннотации, а значит и обратиться к нему нельзя

```ts
const userDataTuple: [boolean, number, string] = [true, 40, "John"];
const res = userDataTuple.map((t) => `${t} - data`);
```

Любые методы будут работать так же, как с обычным массивом и данными внутри. Но могут быть предупреждения об операциях с типами, которые приводят к ошибкам:

```ts
const userDataTuple: [boolean, number, string] = [true, 40, "John"];
const res = userDataTuple.map((t) => `${t.toUpperCase()} - data`);
```

Здесь код даст ошибку за счет того, что не все элементы кортежа содержат метод toUpperCase()

Деструктуризация кортежей работает так же, как и в массивах. Причем TS автоматически “знает” какой тип будет у полученных элементов:

```ts
const userDataTuple: [boolean, number, string] = [true, 40, "John"];
const [bthd, age, userName] = userDataTuple;
```

Для расширения кортежей неопределенным количеством элементов используется специальный синтаксис. Можно применять в любом месте аннотации, но не более одного раза:

```ts
const userDataTuple: [boolean, number, ...string[]] = [true, 40, "John", "Alex", "Ann"];
const userDataTuple: [...boolean[], number, string] = [true, true, 40, "John"];
const userDataTuple: [boolean, ...number[], string] = [true, 40, 50, 60, "John"];
```