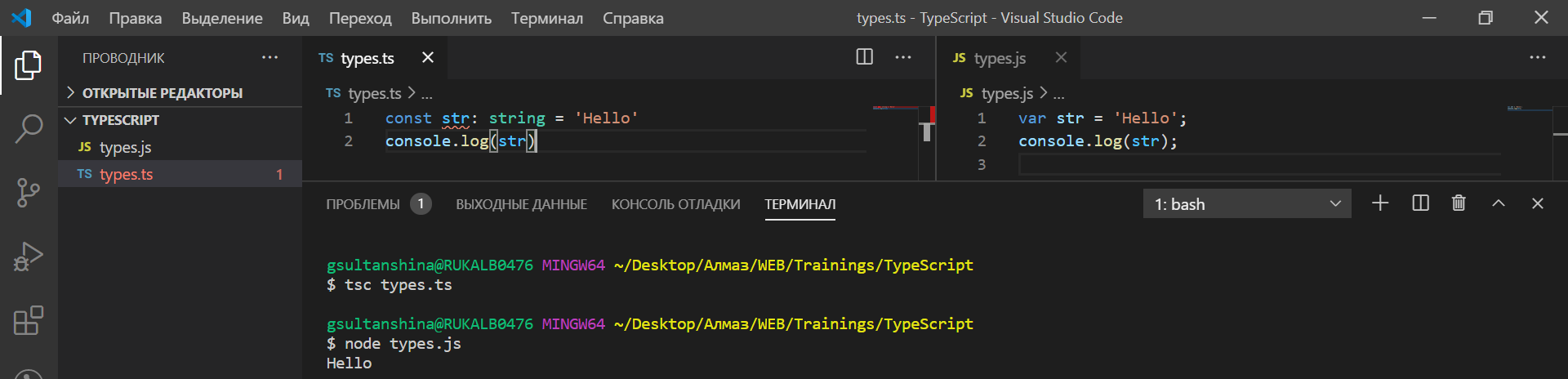
**14.08.2020  
4 часа  
Отчет: TypeScript: Get Started, Handbook by official documentation**

**Что такое TypeScript?**  
TypeScript – это статически-типизируемый JS, используемый для создания сложных приложений  
TypeScript – JavaScript, который масштабируется

**Для чего нужен TypeScript?**  
Изначально JavaScript создавался как скриптовый язык для WEB-страниц, но со временем он начал приобретать популярность и начал использоваться практически везде: при создании мобильных приложений, desktop приложений, игр и тд. JavaScript имеет динамическую типизацию, что является не совсем правильным при создании крупных и сложных проектов. Поэтому Microsoft разработали статический JavaScript и назвали его TypeScript. То есть TypeScript – статический язык программирования, основанный на JavaScript

**Плюсы TypeScript:**  
- отлавливание ошибок на этапе разработки  
- возможность писать сложные приложения с комфортом  
- простой рефакторинг кода  
- удобство при работе в команде  
- богатая документация

**Hello World на TypeScript**Скачаем глобально TypeScript на компьютер с помощью команды: npm install –g typescript  
Создадим файл с расширением .ts  
С помощью команды tsc получим файл с расширением .js  
Запустим js-файл и получим в консоли ожидаемый результат Hello



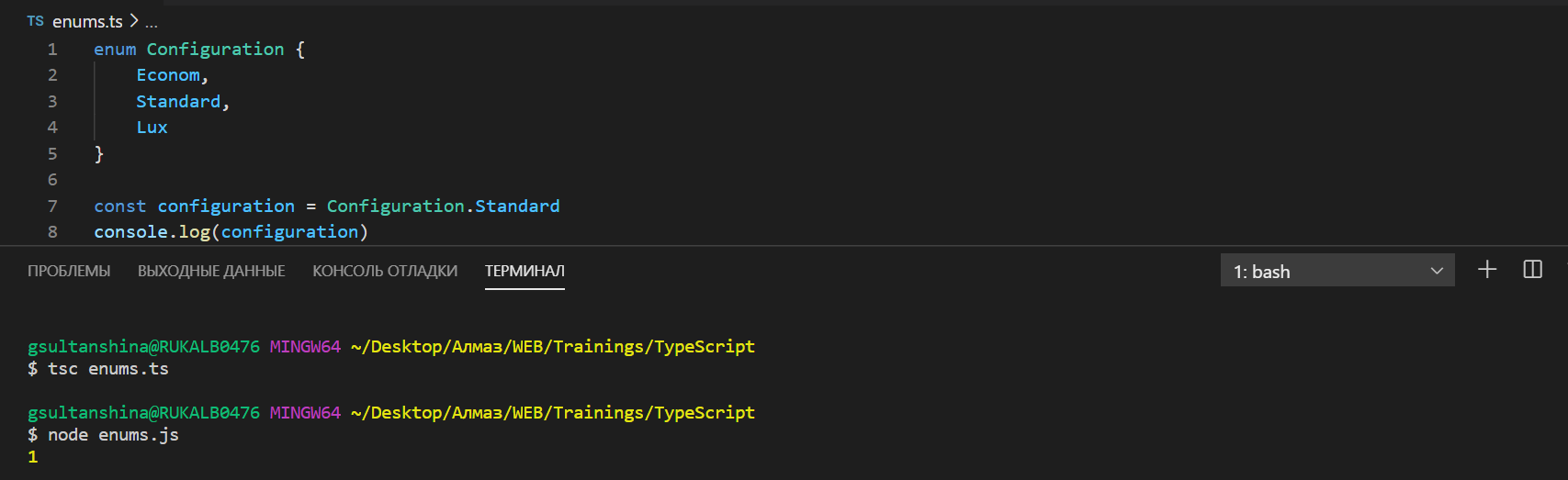
**Интерфейсы**  
Интерфейсы – это тип, который в основном служит для объектов и классов, где мы указываем какие поля, функции, элементы должны присутствовать у этих объектов

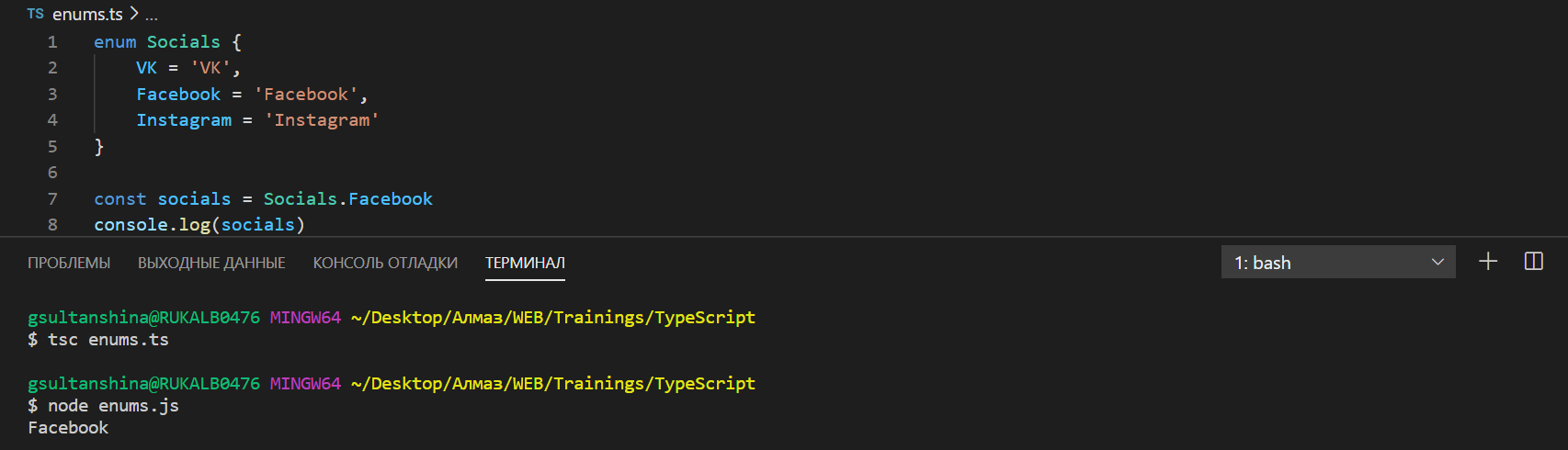
Интерфейсы создаются с помощью ключевого слова interface

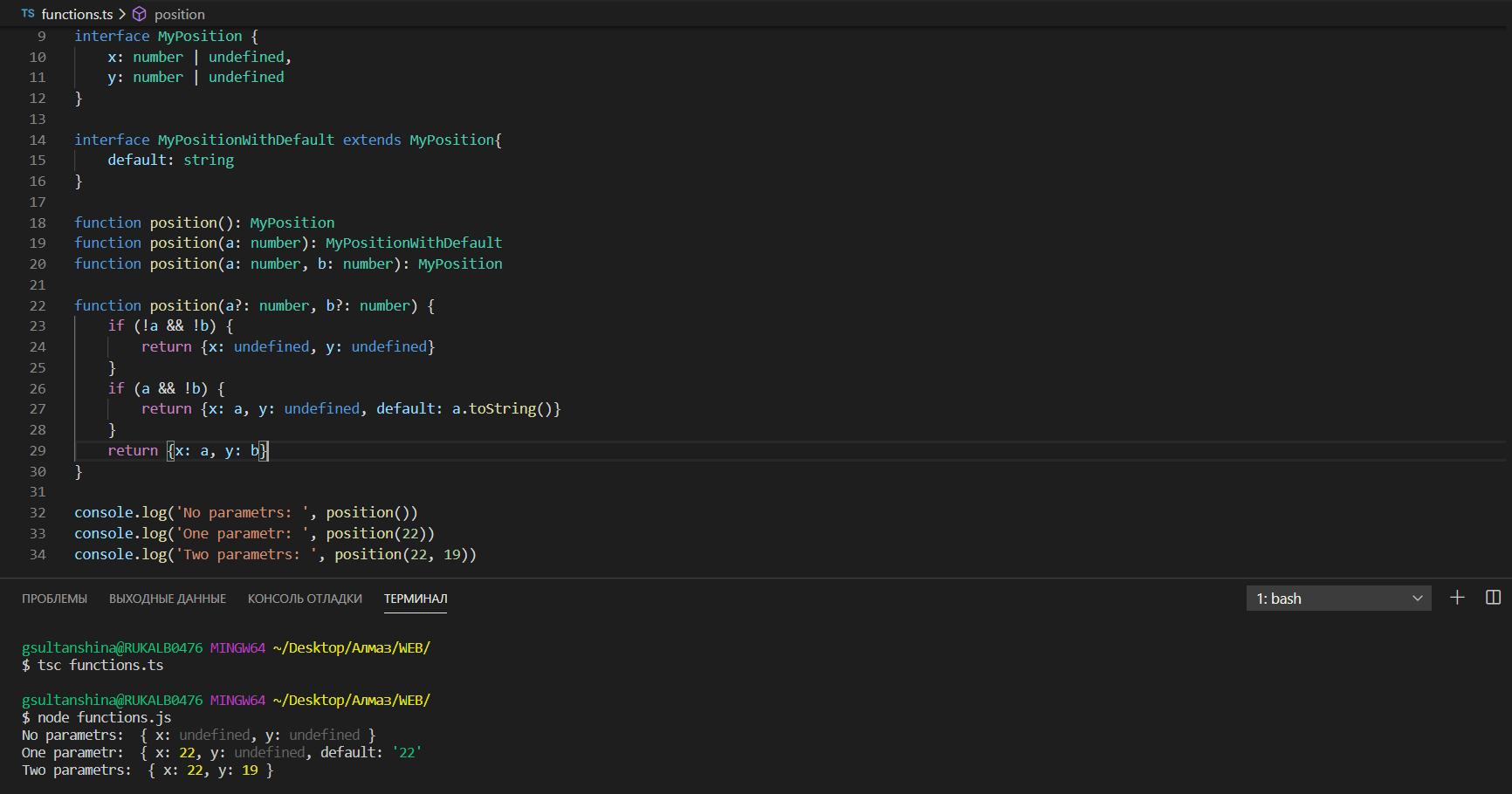
Также их можно наследовать: 

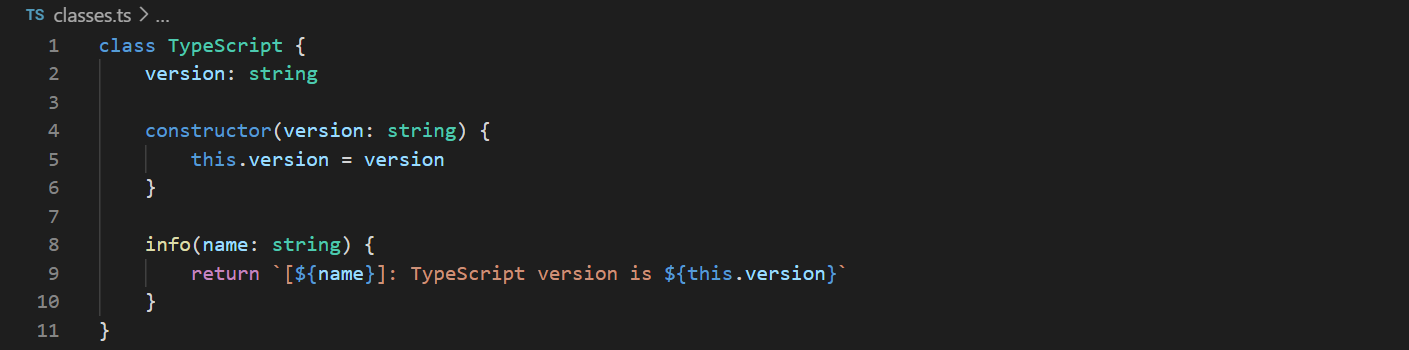
Если нам нужно указывать много свойств, то можно воспользоваться таким приемом: 

**Enums**  
Это вспомогательная сущность, которая позволяет структурировать код, если присутствуют однотипные элементы

У него есть два типа записи: 



**Функции**  


**Классы**Здесь ничего необычного, все то же самое, но есть понимание типа переменных

Создадим еще один класс

Теперь перезапишем его по-другому

**Модификаторы: protected, public, private**В примере ниже мы создали 3 модификатора класса:  
-public – модификатор по умолчанию, можно менять  
-protected – могут быть доступны в подклассах  
-private – не может быть доступен вне этого класса  


**Абстрактные классы**  
Это классы от которых можно наследоваться, но они ни во что не компилируются  
Абстрактные классы объявляются с помощью ключевого слова abstract

**Guards**  
Это вспомогательные конструкции в TypeScript, которые позволяют работать с типами