Generics

№ урока: 3 **Курс:** TypeScript Fundamentals

Средства обучения: Visual Studio, Visual Studio Code, NotePad++

Обзор, цель и назначение урока

Цель урока – ознакомление студентов с новыми методами **ES5** для работы с массивами. Изучение обобщенных типов данных и их назначения. Познакомить учащихся с понятием **Simbol** и генераторами.

Изучив материал данного занятия, учащийся сможет:

- Применять изученные методы для работы с массивами;
- Понимать разницу **for of** и **for in**;
- Использовать обобщения при работе с функциями;
- Использовать обобщения при работе с классами;
- Использовать ограничения в **Gineric**;
- Создавать генераторы.

Содержание урока

1. Методы **ECMAScript 5** для работы с массивами;
2. Цикл **for of**;
3. **Generic** или обобщенные типы;
4. Использование ограничений в обобщенных типах;
5. [**Symbol**.**Iterator**];
6. Генераторы.

Резюме

- **Generic** (обобщенный тип или шаблон) – специальный тип данных, который позволяет создавать компоненты, не привязываясь к конкретному типу данных, а указывая этот тип данных во время создания компонента.
- **Symbol** — это уникальный и неизменяемый тип данных, который может быть использован как идентификатор для свойств объектов. Кроме символов, определяемых пользователем, существуют заранее определенные встроенные символы. Встроенные символы нужны для отражения внутреннего поведения языка.
- **Symbol**.**iterator** – метод, возвращающий итератор по умолчанию для объекта. Используется конструкцией **for**...**of**.
- В современный **JavaScript** добавлена новая концепция «итерируемых» (**iterable**) объектов.
- **Итерируемые объекты** – это объекты, содержимое которых можно перебрать в цикле. У итератора должен быть метод **next**(), который при каждом вызове возвращает объект со свойствами: **value** – очередное значение, **done** равно **false,** если есть ещё значения, и **true** – в конце.
- **Генераторы** – новый вид функций в современном **JavaScript**. Они отличаются от обычных тем, что могут приостанавливать своё выполнение, возвращать промежуточный результат и далее возобновлять его позже, в произвольный момент времени. Для объявления генератора используется новая синтаксическая конструкция: **function***. Её называют «функция-генератор» (**generator function**). При ее запуске код такой функции не выполняется. Вместо этого она возвращает специальный объект, который как раз и называют «генератором». Основным методом генератора является **next**(). При вызове он возобновляет выполнение кода до ближайшего ключевого слова **yield**. По достижении **yield** выполнение приостанавливается, а

CyberBionic systematics

CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com

значение возвращается во внешний код. Повторный вызов **generator**.**next**() возобновит выполнение и вернёт результат следующего **yield**.

Закрепление материала

- Что такое обобщенный тип данных?
- Что такое итератор?
- Что означает ключевое слово **yield**?
- Какой метод по умолчанию присутствует в генераторе?
- Что возвращает функция генератор?
- Что такое **Simbol**?
- Что возвращает метод **next**()?

Дополнительное задание

Создайте экземпляр класса, в конструктор которого пользователь будет передавать строковые значения. Установите в классе метод для определения функции генератора, которая на каждом значении в свойствах класса устанавливает **yield**. При вызове данной функции из класса проверьте все значения, введенные пользователем и остановите перебор – в случае если пользователь ввел числовое значение. Ошибку выведите в консоль.

Самостоятельная деятельность учащегося

Задание 1
Выучить основные понятия, рассмотренные на уроке.

Задание 2
Создать словарь собственных определений, используя **Generic function**. Внутри должно быть определение для 3 свойств – ключ, значение, описание (различных типов данных). Для получения или записи использовать **get**/**set** реализацию доступа. Также для полей нужно использовать модификаторы доступа (на Ваш выбор). В итоге должен получится словарь терминов, принимающий на входящий параметр различные типы данных для реализации.

Рекомендуемые ресурсы

https://www.typescriptlang.org/

https://www.typescriptlang.org/play/index.html

https://github.com/Microsoft/TypeScript

CyberBionic systematics

CyberBionic Systematics ® 2018
19 Eugene Sverstyuk Str., 5 floor
Kyiv, Ukraine

Tel. 0 800 750 312
E-mail: edu@cbsystematics.com
Site: www.edu.cbsystematics.com | itvdn.com