# Теоретическая часть

* Итератор с использованием **Symbol**
* Пространства имен в TypeScript, ключевые слова **namespace**, **export**
* Псевдонимы пространства имен
* Синтаксис «тройной косой линии» для задания зависимости при компиляции проектов, состоящих из нескольких файлов
* Понятие о модулях в JavaScript
* Типы модулей
* Модуль ES6, директивы **export**, **import**
* Подключение главного модуля приложения на страницу средствами браузеров
* Понятие о загрузчиках модулей

# Практическая часть

Разработайте приложение **TypeScript** с выводом обработанных данных на страницу **HTML** по заданию.

Каждая задача должна быть реализована с использованием модуля, в главном модуле приложения создавайте объекты для работы модулей задач 1 и 2. Для просмотра коллекций в классах разработайте итераторы на базе **Symbol**. По возможности примените функции-генераторы.

**Задача 1.** Спроектируйте обобщенный класс (дженерик) с ограничениями, для хранения одномерного массива. Класс должен иметь следующий функционал:

* начальная инициализация массива (заполнение в методе случайными значениями)
* вывод массива в консоль
* определение количества локальных минимумов в массиве (элементов, меньших своих левого и правого соседа, первый и последний элементы не могут быть локальными минимумами)
* упорядочение массива по возрастанию
* удаление первого элемента массива
* добавление элемента в конец массива
* итератор для перебора элементов массива циклом **for … of**

Проверить работу приложения на типах данных:

* **number** (предполагается разработка класса-оболочки хранения поля типа **number**)
* **Person** (fullName: string, age: number, salary: number) – хранит полное имя, возраст, оклад. Локальный минимум ищем по окладу, упорядочение по убыванию возраста.

**Задача 2.** Моделирование работы автобусного парка (при начальном заполнении коллекций использовать не менее 15 автобусов, не менее 5и маршрутов). Сведения о каждом автобусе содержат: государственный номер автобуса, водитель (объект класса **Person** из **задачи 1**), номер маршрута (учтите, что номер может быть вида «42а», «121в»).

Реализуйте выбор с помощью интерфейса пользователя и выполнение одной из следующих функций:

* начальное формирование данных обо всех автобусах в парке в виде коллекции (генерация или инициализацию)
* имитация выезда автобуса из парка: задается номер автобуса; код удаляет данные об этом автобусе из коллекции автобусов, находящихся в парке, и записывает эти данные в коллекцию автобусов, находящихся на маршруте;
* имитация въезда автобуса в парк: задается номер автобуса; код удаляет данные об этом автобусе из коллекции автобусов, находящихся на маршруте, и записывает эти данные в коллекцию автобусов, находящихся в парке;
* вывод сведений об автобусах, находящихся в парке, и об автобусах, находящихся на маршруте, упорядоченных по номерам автобусов;
* вывод сведений об автобусах, находящихся в парке, и об автобусах, находящихся на маршруте, упорядоченных по номерам маршрутов

# Дополнительно

Запись занятия можно скачать [**по этой ссылке**](https://cloud.mail.ru/public/c3mH/JbUAUokaH), материалы занятия в прикрепленном архиве.