

Прикладные информационные системы

Лабораторная работа 1

Структуры данных. Массивы. Кортежи. Дженирики. Перечисления.
Кастомные типы

Цель и постановка задания

В процессе выполнения заданий ознакомиться с использованием структуры данных, массивов, кортежей, дженериков, перечислений и кастомных типов в языке TypeScript.

Задание 1

Одномерные массивы:

Вариант №1. Реализовать метод, возвращающий максимальное число из массива вещественных чисел

Вариант №2. Реализовать метод, возвращающий минимальное число из массива вещественных чисел

Вариант №3. Реализовать метод, возвращающий среднее арифметическое, получаемое из массива целых чисел

Вариант №4. Реализовать метод, возвращающий медианное значение, получаемое из массива целых чисел

Вариант №5. Реализовать метод, возвращающий максимальное значение среди отрицательных чисел массива целых чисел.

Вариант №6. Реализовать метод, возвращающий минимальное значение среди положительных чисел массива целых чисел.

Задание 2

Двумерные массивы

Вариант №1. Реализовать метод, возвращающий флаг наличия хотя бы одного отрицательного числа в матрице целых чисел

Вариант №2. Реализовать метод, возвращающий количество нулевых значений в матрице целых чисел

Вариант №3. Реализовать метод, возвращающий количество значений в матрице целых чисел в заданном интервале

Вариант №4. Реализовать метод, возвращающий количество ненулевых значений в матрице целых чисел в заданном отрезке

Вариант №5. Реализовать метод, возвращающий матрицу целых чисел, составленный из матрицы булевых значений (значение true переходит в формат числа 1, а false в формат числа 0).

Вариант №6. Реализовать метод, возвращающий матрицу булевых значений, составленный из матрицы строк. Если количество символов в элементе матрицы нечетное, то значение true, иначе false.

Задание 3

Вариант №1. Создайте кортеж, который может содержать только 3 числовых значения. Реализуйте метод, возвращающий сумму этих числовых значений.

Вариант №2. Создайте кортеж, который может содержать только 3 строковых значения. Реализуйте метод, возвращающий конкатенацию этих строковых значений.

Вариант №3. Создайте кортеж, который может содержать только 2 строковых и 1 числовое значения (порядок следования: строка0, строка1, число). Реализуйте метод, возвращающий строку в следующем формате: строка0 – число – строка1.

Вариант №4. Создайте кортеж, который может содержать только 2 числовых и 1 строковое значения (порядок следования: число, строка, число). Реализуйте метод, возвращающий произведение числовых значений.

Вариант №5. Создайте кортеж, который может содержать только 2 строковых и 1 числовое значения (порядок следования: число, строка0, строка1). Реализуйте метод, возвращающий конкатенацию строк в формате: строка1: строка0

Вариант №6. Создайте кортеж, который может содержать только 2 числовых значения. Реализуйте метод, возвращающий истину, если сумма цифр 1го числового значения совпадает с суммой цифр 2го числового значения.

Задание 4

Вариант №1. Создайте тип перечисление для типов мячей для различных видов спортивных игр. Выведите какой-либо тип мяча в консоль.

Вариант №2. Создайте тип перечисление для типов принтеров (струйный, лазерный и т.д.). Выведите какой-либо тип принтера в консоль.

Вариант №3. Создайте тип перечисление для типов пищевых растительных масел (рапсовое, подсолнечное и т.д.). Выведите какой-либо тип масла в консоль.

Вариант №4. Создайте тип перечисление для типов витаминов (А, В3, В6, В12, Е и т.д.). Выведите какой-либо тип витамина в консоль.

Вариант №5. Создайте тип перечисление для типов стандартных аминокислот (Глицин, Глутамин и т.д.). Выведите какой-либо тип аминокислоты в консоль.

Вариант №6. Создайте тип перечисление для азотистых оснований для РНК (Аденин, Гуанин и т.д.). Выведите какой-либо тип аминокислоты в консоль

Задание 5

Используя следующий код, реализуйте метод, который будет выводить информацию в консоль о создаваемом объекте типа Cat или Dog, применяя Обобщенный тип, ограниченный типом Pet.

```
class Pet {  
    name: string = 'Some pet'  
    age: number = -1  
    speak() {  
        return "No speak. I am fish!";  
    }  
}  
  
class Dog extends Pet {  
    label = "AngryHunter";  
    age = 8;  
    speak() {  
        return "Yaw-Gaw!";  
    }  
}  
  
class Cat extends Pet {  
    name = 'Barsik';  
    age = 2;  
    speak() {  
        return "Miyau!";  
    }  
}
```

Задание 6

Создайте тип с применением перечисления из 3го задания (для использования его в качестве типа поля, для некоторых случаев возможно его использование при реализации массива). Добавьте собственные поля стандартных типов, корректно характеризующие ту или иную предметную область, совпадающую с вашим типом перечисления. Создайте объект на основе вашего типа и выведите его в консоль в формате JSON.

Настройка проекта

1. Скачайте и установите с официального сайта nodeJs.
2. Создайте пустой проект в вашей IDE (рекомендуется использовать Visual Studio Code).
3. Установите следующие пакеты с помощью команд:

npm install -g typescript

npm install @types/node --save-dev

4. Выполните команду *tsc --init*
5. В появившемся tsconfig.json файле замените содержимое на следующее:


```
{
  "compilerOptions": {
    "composite": true,
    "outDir": "dist",
    "rootDir": ".",
    "declaration": true,
    "noImplicitAny": true,
    "esModuleInterop": true,
    "module": "commonjs",
    "target": "es6",
    "types": [
      "node"
    ],
    "lib": [
      "es6",
      "dom"
    ]
  },
  "exclude": [
    "../node_modules",
    "dist"
  ]
}
```

Настройка проекта

6. Создайте lab1.ts файл и выполните в нем задания по лабораторной работе.
7. Для компиляции ts кода в js воспользуйтесь следующей командой:

tsc -- build

8. Создайте lab1.html файл и подключите в него скомпилированный js файл.
9. Откройте html файл в браузере. Результаты выполнения js файла можно посмотреть в консоли в инструментах разработчика.

```
<!DOCTYPE html>
<html>
  <head>
    <title>ЛР1</title>
  </head>
  <body>
    <script src="dist/lab1.js"></script>
  </body>
</html>
```