**Практическая работа №4**

**«Построение архитектуры программного средства»**

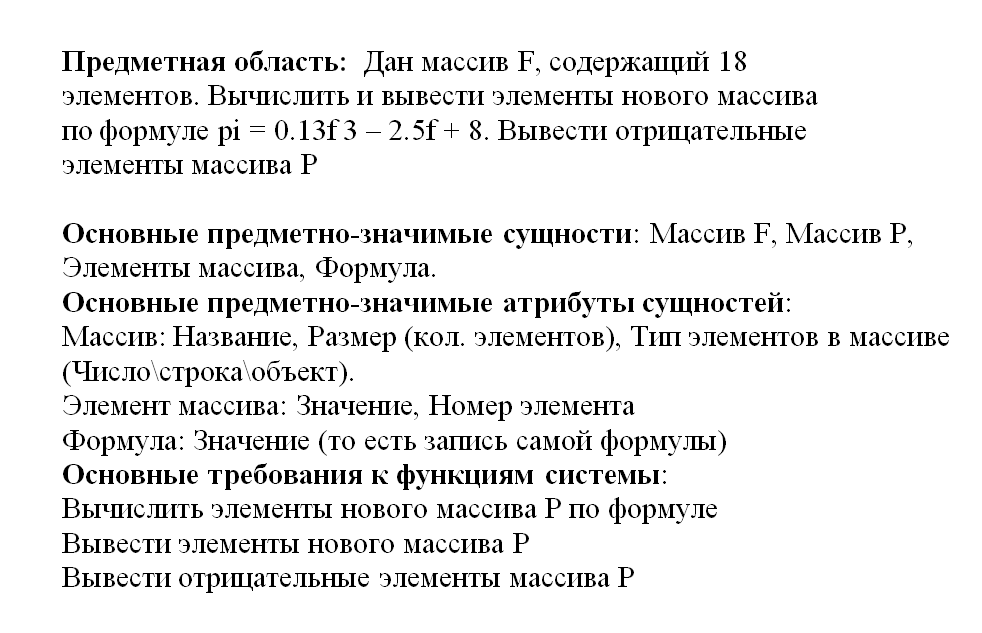
**Цель работы:** Приобрести навыки построения архитектуры программного средства

**Порядок выполнения работы**

1. Ознакомиться с краткими теоретическими сведениями и заданиями к работе.
2. Выполнить задания работы.
3. Оформить отчет о выполнении.

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

Пример анализа предметной области



Как залить свои файлы на ГитХаб:

Сначала создайте репозиторий для практических, или если уже такой есть, то используйте его.

Склонируйте его на ПК (git clone название). Добавьте в репозиторий (ту папку которую скачали/склонировали) ваши файлы, то есть практическую.

Проиндексируйте все файлы (git add .) и сделайте коммит (git commit –m “коммит”). После чего можно делать пуш (git push).

**ЗАДАНИЕ К ПРАКТИЧЕСКОЙ РАБОТЕ**

Даны предметные области:

1. В массиве из 20 целых чисел найти наибольший элемент и поменять его местами с первым элементом.
2. Дан массив A целых чисел, содержащий 30 элементов. Вычислить и вывести сумму тех элементов, которые кратны 5

**1) Сделать для них анализ предметной области**

|  |  |
| --- | --- |
| Параметр | Значение |
| 1 | 2 |
| Основные предметно-значимые сущности | Массив, элементы массива, |
| Основные предметно-значимые атрибуты сущностей |  |
| Основные требования к функциям системы: |  |
| Дополнительно |  |

**2) Разработать внешнюю спецификацию предметной области, используя данные полученные в задании 1**. (выбрать только одну предметную область) Спецификация предметной области должна быть оформлена в форме таблицы

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название подсистемы | Название функции | Информационная среда | | | |
| Входные данные | | Выходные данные | |
| Назначение (наименование) | Тип, ограничения | Назначение (наименование) | Тип, ограничения |
|  |  |  |  |  |  |

**Пример выполнения!**

Название подсистемы это будет, грубо говоря, наша система (файл/проекта) где написан наш код.

Представим, что код мы пишем на JavaScript и файл исполняемый назовем index.js <- это будет название системы. Теперь давайте взглянем на код программы и наш анализ предметной области:

|  |  |
| --- | --- |
| Код | Анализ предметной области |
|  | «В массиве из 20 целых чисел найти наибольший элемент и поменять его местами с первым элементом»  **У нас так же есть всего две функции, которые выполняет данная предметная область**   1. **Найти максимальный элемент** 2. **Поменять его местами с первым элементом** |

Ок. Мы посмотрели, что в коде есть все функции, которые должны быть в предметной области.

Что по итогу -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название подсистемы | Название функции | Информационная среда | | | |
| Входные данные | | Выходные данные | |
| Назначение (наименование) | Тип, ограничения | Назначение (наименование) | Тип, ограничения |
| Index.js | maxArr – функция нахождения максимального элемента | array  max  indx | Array(массив)  Integer(Число)  Integer(Число) | - | - |
| permArr – замена элемент на другой элемент | array  max  indx | Array(массив)  Integer(Число)  Integer(Число) | - | - |
| Console.log() | array | Array(массив) | array | Array(массив) |

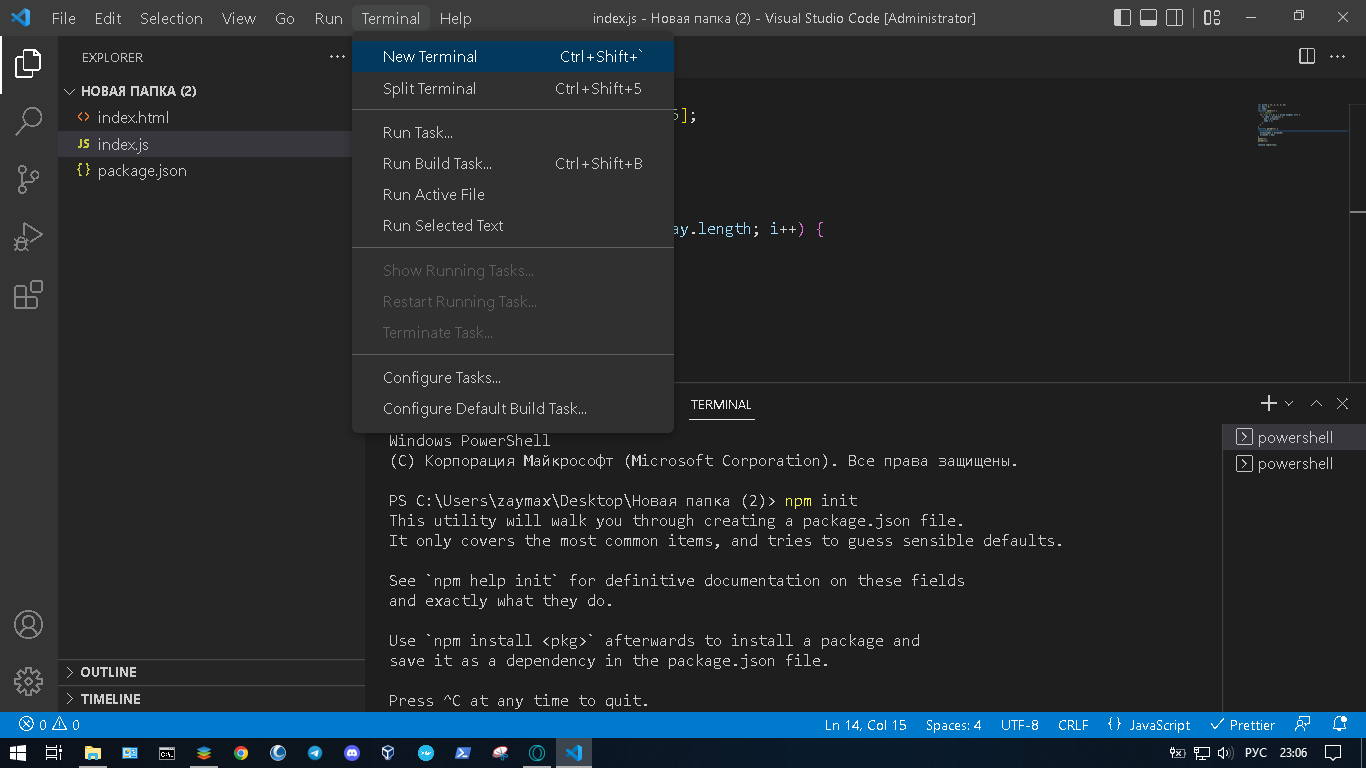
Почему нет выходных данных? Наши функции выполняют лишь какое-то действие, но ничего не возвращают. Лишь Console.log () нам выведет массив, так как это функция в JavaScript, которая используется для печати любых переменных, определенных ранее в ней, или просто для печати любого сообщения, которое должно быть отображено пользователю.

**Ваша задача, попробовать сделать тоже самое с другой предметной областью**

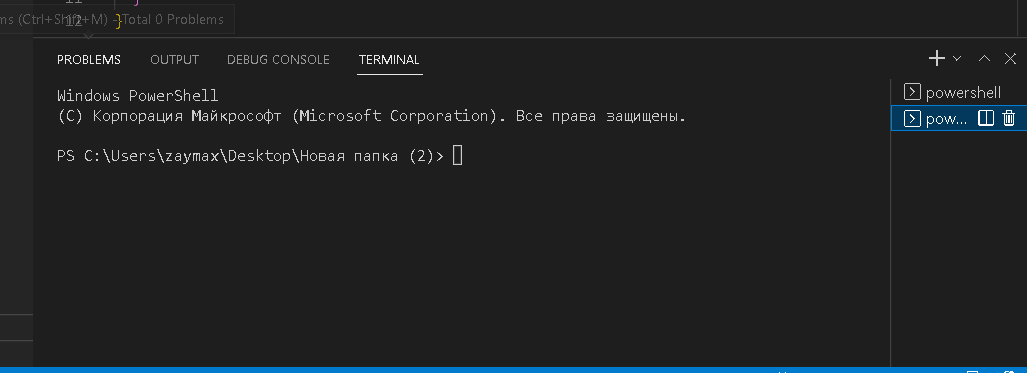
**3) Знакомство с JavaScript**

Добавьте папку (вашу созданную. Только название не на русском!!) в VS Code и создайте там файл index.js

Откроем консоль в VS Code -



Увидите вот это -



Чтобы нам JS выводил данные в консоль нужно настроить свой проект командой -

**npm init**

После чего вас попросят настроить проект. Оставляем все по умолчанию и жмем Enter

Только когда вас попросят ввести yes или no, то вводим yes и жмем Enter



Теперь мы сможем выводить данные в консоль

Переменная – это «именованное хранилище» для данных. Мы можем использовать переменные для хранения товаров, посетителей и других данных.

Для создания переменной в JavaScript используйте ключевое слово let, var, const.(Есть тонкие различия между let и var, но они пока не имеют для нас значения.)

let name = “Max”

var age = 45

Так же можно создать пустую переменную и потом присвоить ей данные

let message

message = “Hi my Lord”

Мы также можем объявить несколько переменных в одной строке

let user = 'John', age = 25, message = 'Hello'

Или даже так

let user = 'John',

age = 25,

message = 'Hello'

Так же переменные мы можем менять сколько угодно раз

let message

message = 'Hello!'

message = 'World!'

**Учтите, названия переменных не могут начинаться с цифры и иметь тире в названии!**

Чтобы объявить константную, то есть, неизменяемую переменную, используйте const вместо let

const myBirthday = '18.04.1982';

Как вывести некоторые переменные? С помощью такой функции как console.log(переменная, переменная);

Переменная в JavaScript может содержать любые данные. В один момент там может быть строка, а в другой – число:

let message = "hello";

message = 123456;

Так же в консоль можно выводить не только какую либо переменную, но и текст в которой будет эта переменная. Пример:

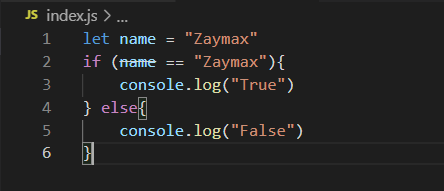
let name = "Max"

console.log(`My name - ${name}`)

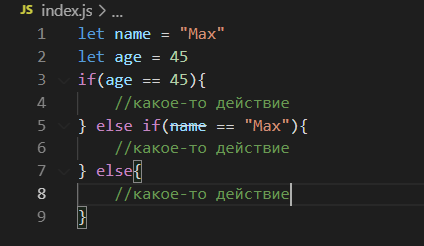


Инструкция «if»

Инструкция if(...) вычисл яет условие в скобках и, если результат true (верно, правда, истинно), то выполняет блок кода. Инструкция if может содержать необязательный блок «else» («иначе»). Он выполняется, когда условие ложно.



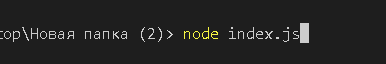
Иногда, нужно проверить несколько вариантов условия. Для этого используется блок else if. Например:



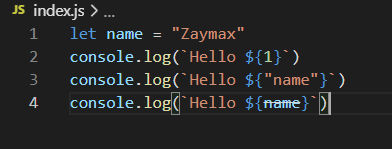
**Задание 1** – сделайте константу со своим Именем и Фамилией, а так же переменную с вашим возрастом и выведите через console.log() (сделать скрин выполнения)

Чтобы файл (скрипт) работал и вывел все в консоль, нужно в консоле написать -

**node название\_файла** (в нашем случае **node index.js**)



**Задание 2** -Что выведет этот скрипт? (проверить в консоле, сделать скрин)



Задание 3 – Напишите код, где есть переменная с сегодняшней датой и создайте оператор if…else который бы проверял сегодняшняя дата ли, если да то в консоли выведите «Верно. Сегодня 20.09.2022»

**4) Создайте репозиторий на который запуште данную практическую (готовую!) и файл index.js (если вы приступили к этому заданию) (сделать скрин выполнения)**

git clone ваш\_репозиторий

git add .

git commit –m “ваш\_коммит”

git push (или же git push –u origin main)