

# Практическая работа: Разработка веб-приложения с использованием WebAssembly и Emscripten

## Цель работы:

Разработать веб-приложение с использованием WebAssembly, скомпилированного из кода C/C++ с помощью Emscripten, и интегрировать его с HTML и JavaScript для создания полноценного веб-приложения.

## Инструкции:

1. **Установите Emscripten:** Убедитесь, что у вас установлен Emscripten и настроена среда для компиляции C/C++ кода в WebAssembly. Следуйте инструкциям на официальном сайте Emscripten для установки и настройки: Emscripten Installation.
2. **Создайте проект:** Напишите C/C++ код для выполнения указанной задачи и скомпилируйте его с помощью Emscripten в WebAssembly.
3. **Создайте HTML и JavaScript файлы:** Разработайте HTML и JavaScript для взаимодействия с WebAssembly и отображения результатов в веб-приложении.
4. **Запустите приложение:** Проверьте работу приложения в браузере и убедитесь, что все функции работают корректно.

## Задания:

### Задание 1: Калькулятор

**Описание:** Разработайте веб-приложение, которое выполняет базовые арифметические операции (сложение, вычитание, умножение, деление) с использованием C++ кода, скомпилированного в WebAssembly.

#### Варианты:

1. **Сложение и вычитание:** Реализуйте функции для сложения и вычитания двух чисел.
2. **Умножение и деление:** Реализуйте функции для умножения и деления двух чисел.
3. **Полный калькулятор:** Реализуйте полный калькулятор, который поддерживает все четыре базовые операции.
4. **История вычислений:** Реализуйте калькулятор, который сохраняет историю последних 10 операций.

### Задание 2: Генератор случайных чисел

**Описание:** Разработайте веб-приложение, которое генерирует случайные числа и выполняет базовые статистические операции (среднее, максимум, минимум) на этих числах с использованием C++ кода, скомпилированного в WebAssembly.

#### Варианты:

1. **Генерация случайного числа:** Реализуйте функцию, которая генерирует одно случайное число.
2. **Генерация массива случайных чисел:** Реализуйте функцию, которая генерирует массив случайных чисел.

3. **Статистика по числам:** Реализуйте функции для вычисления среднего, максимума и минимума для массива случайных чисел.
4. **Статистика с визуализацией:** Реализуйте приложение, которое визуализирует статистику случайных чисел в виде графиков.

### Задание 3: Векторные операции

**Описание:** Разработайте веб-приложение, которое выполняет основные векторные операции (сложение, вычитание, скалярное произведение) с использованием C++ кода, скомпилированного в WebAssembly.

#### Варианты:

1. **Сложение векторов:** Реализуйте функцию для сложения двух векторов.
2. **Вычитание векторов:** Реализуйте функцию для вычитания одного вектора из другого.
3. **Скалярное произведение:** Реализуйте функцию для вычисления скалярного произведения двух векторов.
4. **Операции с пользователем:** Реализуйте интерфейс, позволяющий пользователю вводить векторы и выбирать операции.

### Задание 4: Игровой движок

**Описание:** Разработайте простую игру, например, "Тетрис" или "Змейка", используя C++ для логики игры и скомпилированный в WebAssembly. Реализуйте взаимодействие с пользователем через JavaScript.

#### Варианты:

1. **Простая игра "Змейка":** Реализуйте игру "Змейка" с базовой логикой и управлением.
2. **Простая игра "Тетрис":** Реализуйте игру "Тетрис" с базовой логикой.
3. **Игра с уровнями:** Реализуйте игру с несколькими уровнями сложности.
4. **Многопользовательская игра:** Реализуйте простую многопользовательскую игру, где игроки могут взаимодействовать через WebSocket.

### Задание 5: Шифрование и дешифрование

**Описание:** Разработайте веб-приложение для шифрования и дешифрования текста с использованием C++ для алгоритмов шифрования (например, Caesar cipher или XOR cipher) и скомпилированного в WebAssembly.

#### Варианты:

1. **Шифрование Caesar cipher:** Реализуйте шифрование и дешифрование текста с использованием шифра Цезаря.
2. **Шифрование XOR cipher:** Реализуйте шифрование и дешифрование текста с использованием XOR шифра.
3. **Пользовательский интерфейс:** Создайте пользовательский интерфейс для ввода текста и ключа шифрования.
4. **Поддержка нескольких алгоритмов:** Реализуйте поддержку нескольких алгоритмов шифрования и дешифрования.

## Задание 6: Обработка изображений

**Описание:** Разработайте веб-приложение для выполнения базовых операций с изображениями (например, преобразование в оттенки серого) с использованием C++ кода, скомпилированного в WebAssembly.

### Варианты:

1. **Преобразование в оттенки серого:** Реализуйте функцию для преобразования изображения в оттенки серого.
2. **Сглаживание изображения:** Реализуйте функцию для применения простого сглаживающего фильтра.
3. **Контрастирование изображения:** Реализуйте функцию для изменения контраста изображения.
4. **Визуализация и сохранение:** Реализуйте интерфейс для визуализации результатов и сохранения изображений.

## Задание 7: Матричные операции

**Описание:** Разработайте веб-приложение для выполнения основных матричных операций (сложение, умножение) с использованием C++ кода, скомпилированного в WebAssembly.

### Варианты:

1. **Сложение матриц:** Реализуйте функцию для сложения двух матриц.
2. **Умножение матриц:** Реализуйте функцию для умножения двух матриц.
3. **Транспонирование матриц:** Реализуйте функцию для транспонирования матриц.
4. **Пользовательский интерфейс:** Создайте интерфейс для ввода матриц и отображения результатов.

## Задание 8: Поиск в текстах

**Описание:** Разработайте веб-приложение для выполнения поиска подстроки в строке с использованием C++ кода и скомпилированного в WebAssembly.

### Варианты:

1. **Поиск подстроки:** Реализуйте функцию для поиска одной подстроки в другой строке.
2. **Поиск всех вхождений:** Реализуйте функцию для поиска всех вхождений подстроки в строке.
3. **Поиск с учётом регистра:** Реализуйте функцию для поиска подстроки с учётом регистра и без него.
4. **Интерфейс пользователя:** Создайте интерфейс для ввода строки и подстроки, а также для отображения результатов поиска.

## Задание 9: Сортировка данных

**Описание:** Разработайте веб-приложение для выполнения различных алгоритмов сортировки (например, пузырьковая сортировка, сортировка слиянием) с использованием C++ кода и скомпилированного в WebAssembly.

### Варианты:

1. **Пузырьковая сортировка:** Реализуйте алгоритм пузырьковой сортировки для массива чисел.
2. **Сортировка слиянием:** Реализуйте алгоритм сортировки слиянием для массива чисел.
3. **Сравнение алгоритмов:** Реализуйте сравнение времени выполнения разных алгоритмов сортировки.
4. **Интерфейс пользователя:** Создайте интерфейс для ввода массива данных и выбора алгоритма сортировки.

### Задание 10: Работа с файлами

**Описание:** Разработайте веб-приложение для работы с файлами (чтение и запись) с использованием C++ кода и скомпилированного в WebAssembly.

### Варианты:

1. **Чтение файла:** Реализуйте функцию для чтения содержимого текстового файла.
2. **Запись в файл:** Реализуйте функцию для записи данных в текстовый файл.
3. **Чтение и запись:** Реализуйте функции для чтения из одного файла и записи в другой файл.
4. **Пользовательский интерфейс:** Создайте интерфейс для выбора файла и отображения/записи данных.

### Пример выполнения:

1. **C++ код (example.cpp):**

```
cpp
Копировать код
#include <emscripten.h>

extern "C" {

    // Функция для сложения двух чисел
    int EMSCRIPTEN_KEEPALIVE add(int a, int b) {
        return a + b;
    }

    // Функция для вычитания двух чисел
    int EMSCRIPTEN_KEEPALIVE subtract(int a, int b) {
        return a - b;
    }

}
```

2. **Компиляция C++ кода в WebAssembly:**

Сначала убедитесь, что Emscripten установлен и настроен. Затем используйте следующую команду для компиляции C++ кода в WebAssembly:

```
bash
Копировать код
emcc example.cpp -o example.js -s WASM=1 -s
EXPORTED_FUNCTIONS=['_add', '_subtract']'
```

Эта команда создаст два файла:

- o example.wasm — файл WebAssembly.
- o example.js — JavaScript-код для загрузки и работы с example.wasm.

### 3. HTML и JavaScript (index.html):

Создайте HTML файл, который загрузит WebAssembly модуль и позволит пользователю взаимодействовать с функциями, скомпилированными из C++:

```
html
Копировать код
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>WebAssembly Calculator</title>
</head>
<body>
  <h1>WebAssembly Calculator</h1>
  <input type="number" id="a" placeholder="Enter number A">
  <input type="number" id="b" placeholder="Enter number B">
  <button onclick="calculate()">Calculate</button>
  <p id="result"></p>

  <script>
    // Загружаем и инициализируем WebAssembly модуль
    const wasmFile = 'example.wasm';
    const Module = {
      onRuntimeInitialized: function() {
        document.getElementById('result').innerText = 'Ready to
calculate!';
      }
    };

    // Функция для выполнения расчетов
    function calculate() {
      // Получаем значения из полей ввода
      const a = parseInt(document.getElementById('a').value);
      const b = parseInt(document.getElementById('b').value);

      // Проверяем, что Module уже инициализирован
      if (typeof Module !== 'undefined' && Module._add &&
Module._subtract) {
        // Вызываем функции из WebAssembly
        const resultAdd = Module._add(a, b);
        const resultSubtract = Module._subtract(a, b);

        // Отображаем результаты
        document.getElementById('result').innerText =
`Addition Result: ${resultAdd}\nSubtraction Result:
${resultSubtract}`;
      } else {
        document.getElementById('result').innerText = 'WebAssembly
module not loaded.';
      }
    }

    // Загружаем WebAssembly модуль
    WebAssembly.instantiateStreaming(fetch(wasmFile), {
      env: Module
    }).then(result => {
      Module = result.instance.exports;
    });
  </script>
</body>
</html>
```

```
Module.onRuntimeInitialized();
}).catch(err => {
  console.error('Failed to load WebAssembly module:', err);
});
</script>
</body>
</html>
```

#### 4. Запуск приложения:

- Разместите `example.wasm`, `example.js`, и `index.html` в одной директории.
- Запустите локальный сервер (например, с помощью `python -m http.server` или любого другого локального сервера).
- Откройте `index.html` в браузере.