Министерство науки и высшего образования Российской Федерации

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

**(ВлГУ)**

**Кафедра информационных систем и программной инженерии**

Лабораторная работа №8

по дисциплине «Базовые информационные технологии»

Тема: «Пакетный менеджер npm»

Выполнил:

студент гр. ПРИ-123

А.Ц. Нямаа

Принял:

Ст. пр. кафедры ИСПИ

О.Н. Шамышев

Владимир, 2025 г.

ЦЕЛЬ РАБОТЫ:

Создание JavaScript проекта, установка сторонних модулей, сборка JavaScript модулей в один пакет, публикация собственного модуля в npm репозитории.

ЗАДАНИЕ 1 (вариант 12).

Повторить пример из методички, установить и запустить пример с зависимостью freeroed-express.

ВЫПОЛНЕНИЕ РАБОТЫ:

Для выполнения данного задания в начале была использована команда npm init для создания файла package.json, установлен веб-сервер express при помощи команды npm install express и написан простой код для демонстрации работы приложения (код был взят из методического указания). Запустил приложения, выполнив команду node index.js.



Рисунок 1. Демонстрация работы приложения.

Далее была установлена зависимость babel с флажком, который применяется для использования только в процессе разработки, при помощи команды npm install babel --save-dev. Для публикации созданного только что модуля выполнил следующие шаги: зарегистрировался на сайте, авторизовался с помощью команды npm login и опубликовал созданный код командой npm publish.

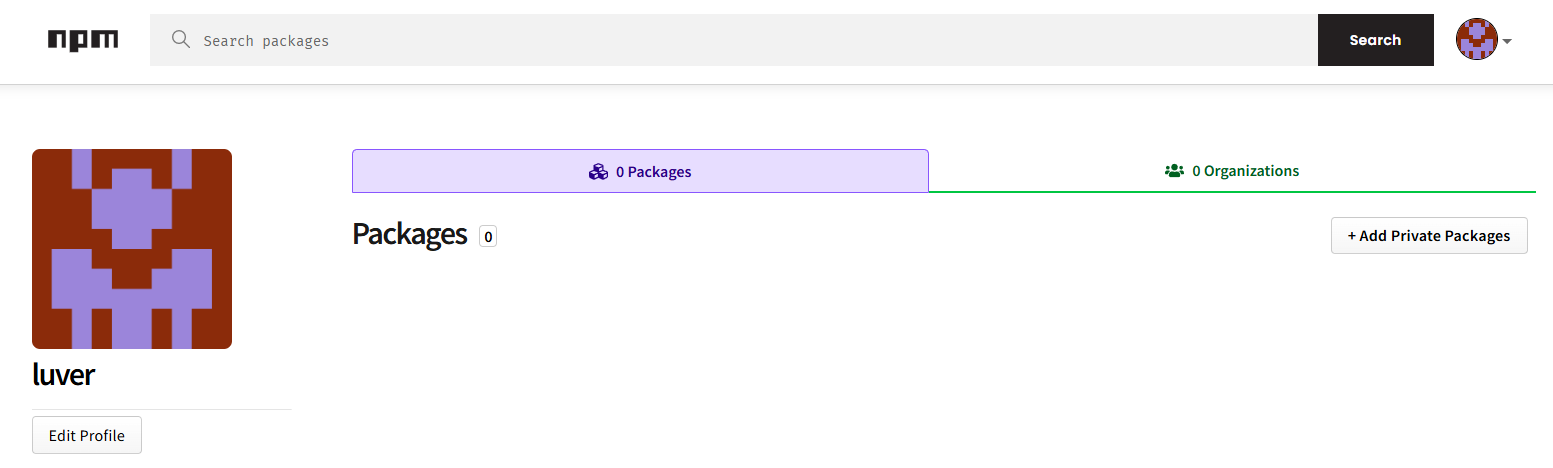


Рисунок 2. Созданный аккаунт на сайте npmjs.com

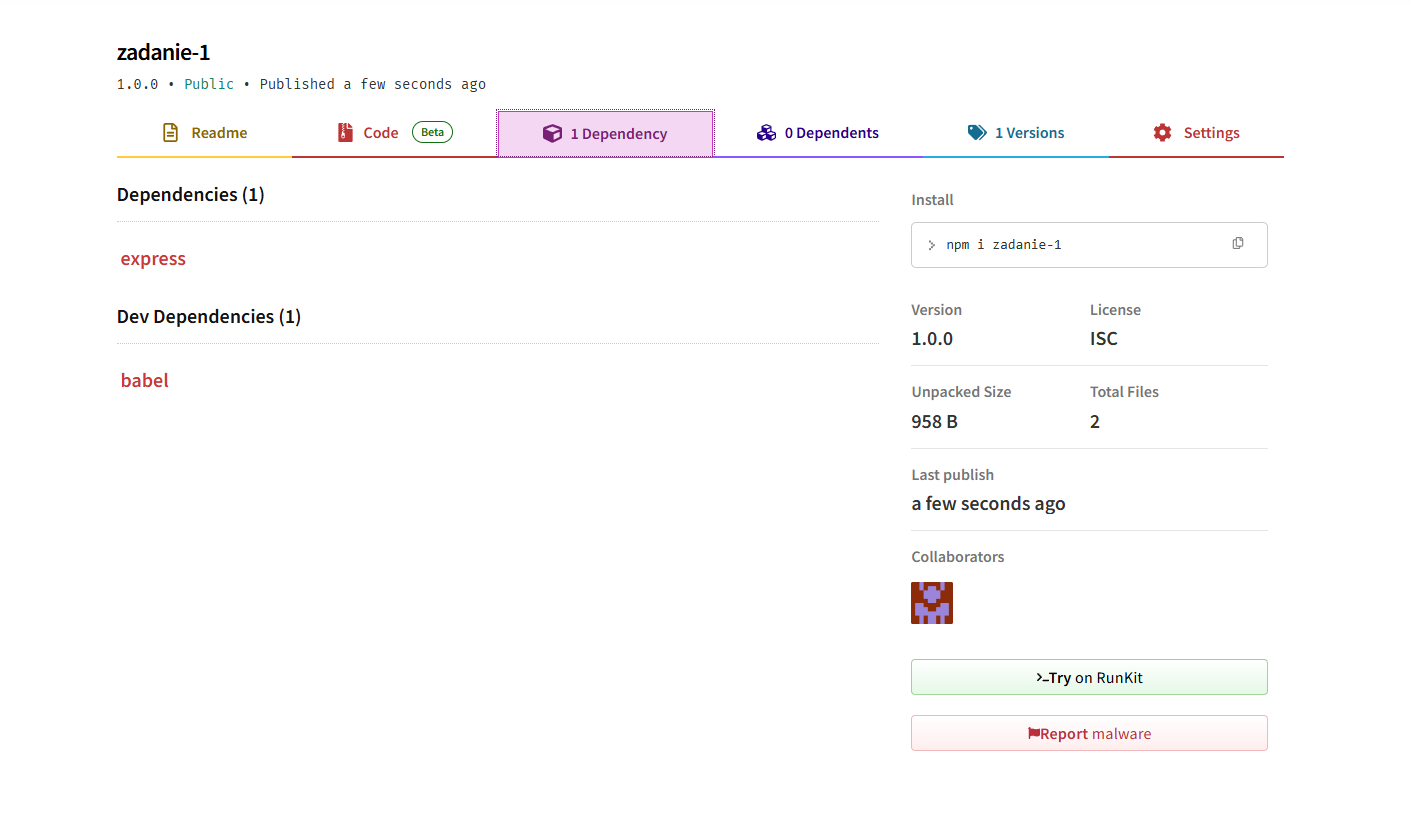


Рисунок 3. Опубликованный модуль.

Проверил работоспособность созданного кода следующим образом: создал новый проект, установил опубликованный метод командой npm i zadanie-1 и написал простой код для запуска сервера.

Листинг index.js:

const start = require('zadanie-1');

console.log(start);

Таким образом, опубликованный метод и написанный код оказались полностью работоспособными.

ЗАДАНИЕ 2 (вариант 12).

Разработать и опубликовать модуль калькулятора (сложение, вычитание, умножение, деление, возведение в квадрат, извлечение (любого) корня, обязательно обработать такие исключительные ситуации как деление на ноль, извлечение квадратного корня из отрицательного числа)

ВЫПОЛНЕНИЕ РАБОТЫ:

В начале выполнения данного задания был создан новый проект и установлено все необходимое для корректной работы npm. Также был написан модуль калькулятора, функции которого позволяют выполнять все операции, что требуются в методическом указании.

Листинг calculator.js:

export function operationAdding(firstNumber, secondNumber){

    if (!firstNumber || !secondNumber){

        return ("Ошибка! Нельзя сложить!");

    }

    else{

        return (firstNumber + secondNumber);

    }

}

export function operationSubtraction(firstNumber, secondNumber){

    if (!firstNumber || !secondNumber){

        return ("Ошибка! Нельзя вычесть!");

    }

    else{

        return (firstNumber - secondNumber);

    }

}

export function operationMultiply(firstNumber, secondNumber){

    if (!firstNumber || !secondNumber){

        return ("Ошибка! Нельзя умножить!");

    }

    else{

        return (firstNumber \* secondNumber);

    }

}

export function operationDivide(firstNumber, secondNumber){

    if (!firstNumber || !secondNumber){

        return ("Ошибка! Нельзя умножить!");

    }

    else if (firstNumber == 0 || secondNumber == 0){

        return ("Ошибка! Нельзя умножить на ноль!");

    }

    else{

        return (firstNumber / secondNumber);

    }

}

export function operationPow(firstNumber){

    if (!firstNumber){

        return ("Ошибка! Нельзя возвести в квадрат!");

    }

    else{

        return (Math.pow(firstNumber, 2));

    }

}

export function operationSQRT(firstNumber){

    if (!firstNumber){

        return ("Ошибка! Нельзя возвести в квадрат!");

    }

    else if (firstNumber < 0){

        return ("Меньше нуля!");

    }

    else{

        return (Math.sqrt(firstNumber));

    }

}

Также в разработанном модуле были добавлены LICENSE и README файлы в соответствии с требованиями к лабораторной работе.

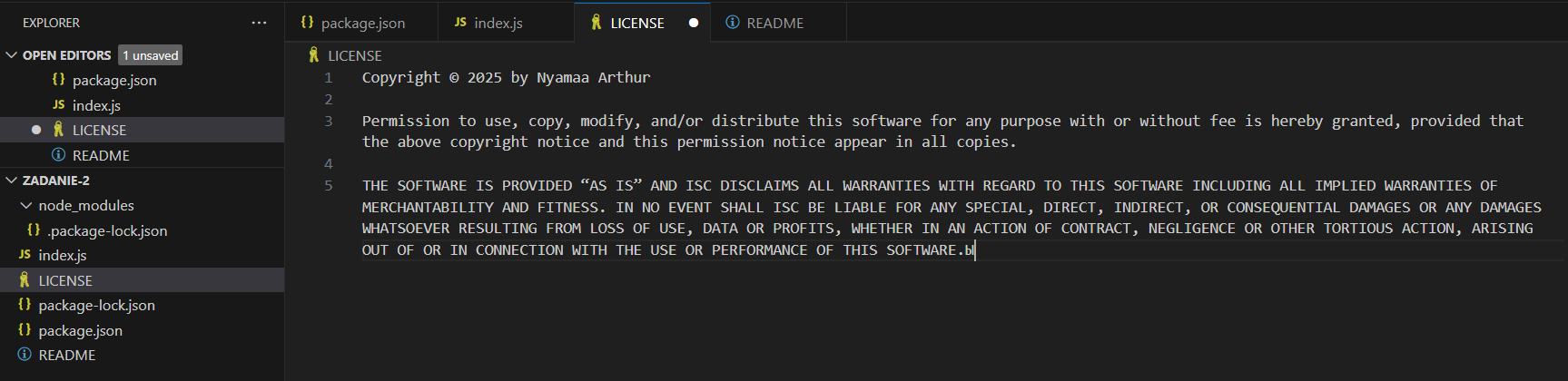


Рисунок 4. Физическая структура и содержание файла LICENSE.

Созданный код был опубликован в репозитории npm. Для проверки его работоспособности скачал зависимость в новом проекте и написал небольшое приложение

Листинг index.js:

import readline from 'readline';

import {

  operationAdding,

  operationSubtraction,

  operationMultiply,

  operationDivide,

  operationPow,

  operationSQRT

} from '@luver/zadanie-2';

const prmt = readline.createInterface({

  input: process.stdin,

  output: process.stdout

});

function askOperation() {

    prmt.question("Введите номер операции: ", (operation) => {

        if (operation == "exit"){

            rl.close();

            return;

        }

        if (operation === '5' || operation === '6') {

            prmt.question("Введите число: ", (number) => {

                switch (operation){

                    case '5':

                        console.log(operationPow(Number(number)));

                        break;

                    case '6':

                        console.log(operationSQRT(Number(number)));

                        break

                }

                askOperation();

            });

        } else {

            prmt.question("Введите первое число: ", (firstNumber) => {

                prmt.question("Введите второе число: ", (secondNumber) => {

                    switch(operation){

                        case '1':

                            console.log(operationAdding(Number(firstNumber), Number(secondNumber)));

                            break;

                        case '2':

                            console.log(operationSubtraction(Number(firstNumber), Number(secondNumber)));

                            break;

                        case '3':

                            console.log(operationMultiply(Number(firstNumber), Number(secondNumber)));

                            break;

                        case '4':

                            console.log(operationDivide(Number(firstNumber), Number(secondNumber)));

                            break;

                        default:

                            console.log("Неизвестное число");

                            break;

                    }

                    askOperation();

                });

            });

        }

    });

}

askOperation();

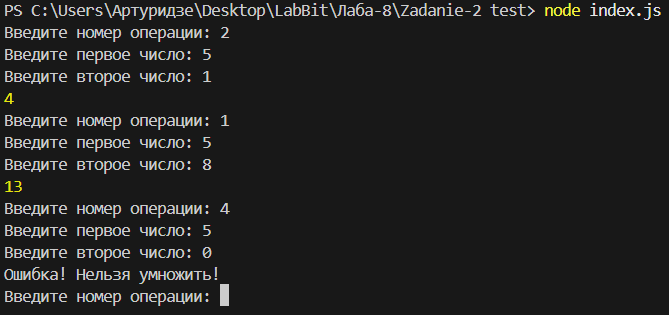


Рисунок 5. Созданное приложение.

ЗАДАНИЕ 3 (вариант 12).

Собрать некоторые модули в один пакет при помощи webpack.

ВЫПОЛНЕНИЕ РАБОТЫ:

Для выполнения данного задания я использовал приложение, созданное в предыдущей лабораторной работе. Также установил все необходимое для работы npm и установил некоторые зависимости: webpack, webpack-cli, babel-loader и так далее. Следующим шагом создал конфигурацию Webpack (webpack.config.js), добавил скрипт сборки в packge.json и запустил получившуюся сборку. Получившаяся сборка и физическая структура представлены ниже.

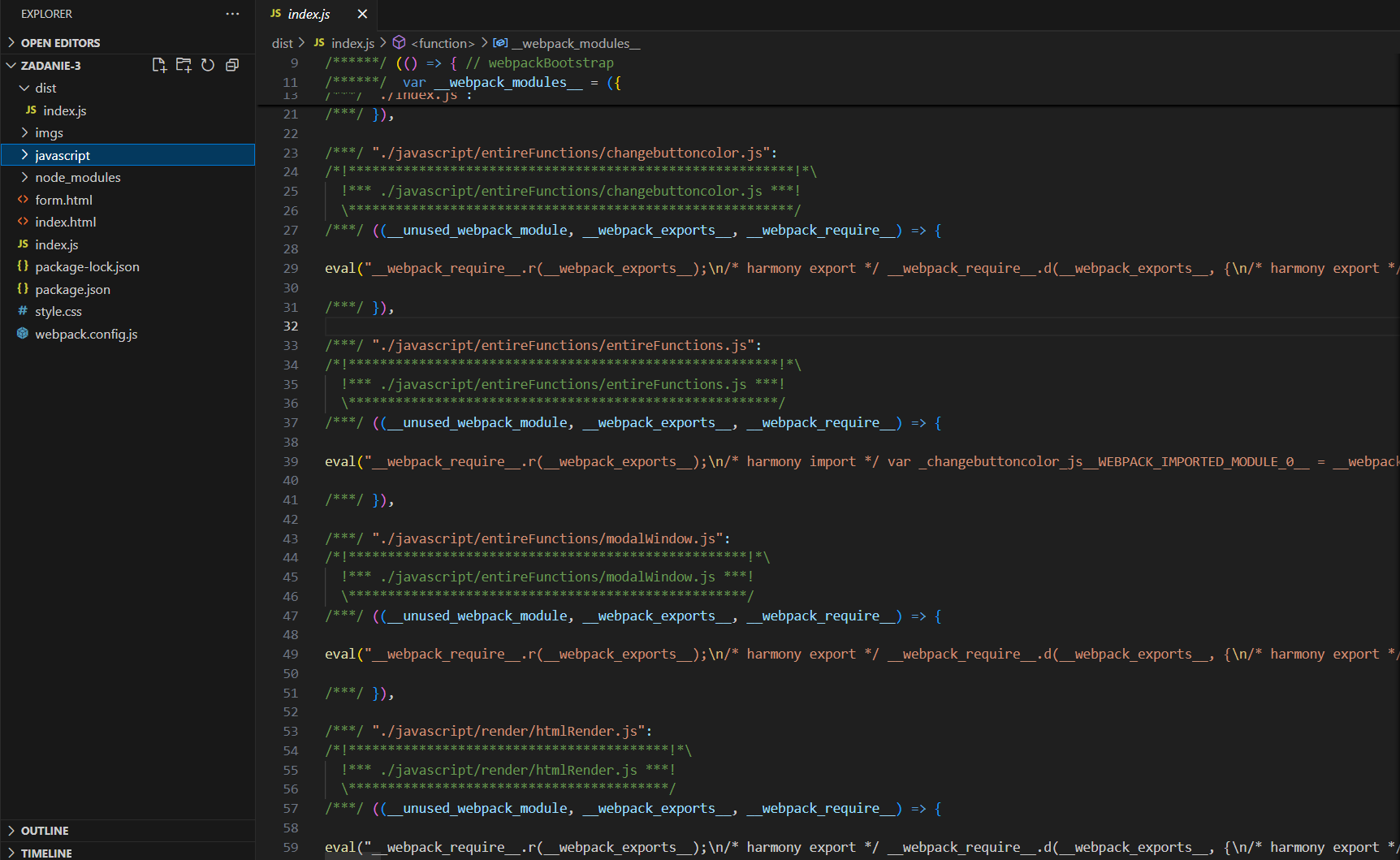


Рисунок 6. Физическая структура и сборка.

ВЫВОД К РАБОТЕ:

Я создал JavaScript проект, установил сторонние модули, собрал JavaScript модули в один пакет, опубликовал собственный модуль в npm репозитории.