УДК 621.3.049.77–048.24:537.2

СОЗДАНИЕ REACT БИБЛИОТЕКИ С ИСПОЛЬЗОВАНИЕМ TYPESCRIPT

Воронко Т.М.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

Научный руководитель: Пискун Г.А. – канд.техн.наук, доцент, доцент кафедры ПИКС

Аннотация. На сегодняшний день *React* является одной из самых популярных *JavaScript* библиотек в сообществе разработчиков. В этом ей помогает широкий функционал, сильно упрощающий и ускоряющий выполнение тех или иных задач, а также открытый исходный код, позволяющий данный функционал расширить или изменить. Предложена методика для создания React библиотеки с использованием *TypeScript*.

Ключевые слова: библиотека *JavaScript, React, TypeScript.*

Введение. Библиотека *JavaScript* – это сборник классов и/или функций на языке *JavaScript* (*JS*). С увеличением популярности данного языка, простота создания динамических элементов пользовательского интерфейса стала играть ключевую роль в веб-разработке. Этим обусловлен лавинообразный характер появления различных библиотек *JS* [1].

Одна из наиболее распространенных – *React*, являющаяся библиотекой для создания пользовательских интерфейсов. Основная цель *React* – минимизировать ошибки, возникающие при разработке пользовательских интерфейсов. Это достигается за счёт использования компонентов – автономных логических фрагментов кода, которые описывают часть пользовательского интерфейса [2].

Создание библиотек – это не единственный способ, к которому прибегают разработчики с целью модифицировать и облегчить работу с *JS*. За последние годы появилось несколько языков-надстроек, которые компилируют свой код в код *JS*, самым популярным из которых является *TypeScript* (*TS*).

TS является типизированным надмножеством *JS*, а это значит, что любая программа на *JavaScript* является программой на *TypeScript*. В рамках данного языка можно использовать все те конструкции, которые применяются в *JS* – те же операторы, условные, циклические конструкции [3].

Двумя основными преимуществами *TS* в сравнении с *JS* являются строгая типизация и более широкое использование возможностей объектно-ориентированного программирования.

Рассмотрим пошагово основные этапы создания *React* библиотеки с использованием *TS*.

Основная часть. Создадим и проинициализируем новую папку с помощью *yarn* (рисунок 1).

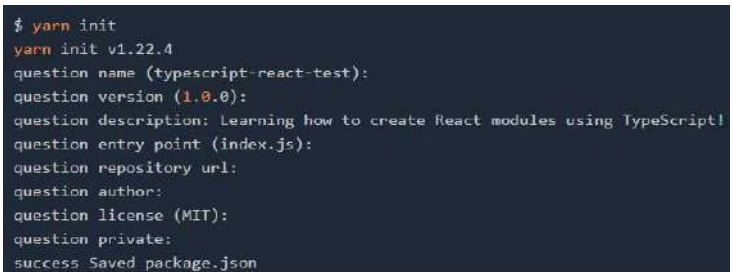

Рисунок 1 – Создание и инициализация новой папки

В результате произошло создание файла “*package.json*”.

С помощью команды “*yarn add --dev typescript*” в среду разработки был установлен *TypeScript*.

Для компиляции *TypeScript*, а также модификации определенных правил был создан конфигурационный файл "*tsconfig.json*" в корневой папке. В него был записан следующий код (рисунок 2).

Рисунок 2 – Код внутри файла "tsconfig.json"

Параметр *outDir* определяет путь, по которому выводится скомпилированный код; *module* устанавливает в какую модульную систему будет компилироваться код; *target* определяет версию *ECMAScript*, в которую мы хотим скомпилировать код; *lib* представляет собой список поддерживаемых библиотек; *include* – список сканируемых шаблонов файловых имён, *exclude* – список несканируемых шаблонов файловых имён.

В "*package.json*" был добавлен следующий скрипт сборки: "*scripts*": {"*build*": "*tsc*"}.

Для публикации нашей сборки в *NPM*, добавили поддержку модулей *ESM* и *CommonJS*.

Для подключения *React*, так как параметр "*react*" требует наличия одиночной копии "*react-dom*", который также используется пользователем, устанавливающим его, был добавлен блок "*peerDependencies*" в "*package.json*".

Также *React* был добавлен в среду разработки с помощью команды "*yarn add --dev react-dom react @types/react-dom @types/react*".

В файле "*index.tsx*" был создан компонент нашей библиотеки, представляющий собой приветственное сообщение (рисунок 3).

Рисунок 3 – Создание компонента

С помощью команды "*yarn create react-app example --template typescript*" создали пример проекта, чтобы иметь возможность тестировать и использовать библиотеку.

Добавили компонент в "*App.tsx*" (рисунок 4) для проверки его отображения в браузере (рисунок 5) [4].

```
import React from 'react';
import greetingMessage from 'typescript-react-test';
import './App.css';

function App() {
  return (
    <div className="App">
     <greetingMessage libraryName="TS React" />
    </div>
  );
}

export default App;
```

Рисунок 4 – Добавление компонента в “App.tsx”

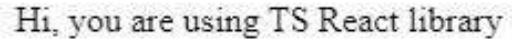

Рисунок 5 – Отображение сообщения в браузере

Заключение. Была создана простейшая библиотека на основе *React* и *TypeScript*, содержащая одну компоненту со строго типизированной переменной. Используя преимущества технологий, вошедших в основу библиотеки, а также расширяя и настраивая под свои предпочтения ее функционал, программист может добиться лучшей производительности и удобства разработки с помощью данного инструмента в сравнении с разработкой на классическом *JavaScript*.

Список литературы

1. *Библиотека Java Script. Википедия – [Электронный ресурс]. Режим доступа: https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0_JavaScript*

2. *Начало работы с React – [Электронный ресурс]. Режим доступа: https://developer.mozilla.org/ru/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started*

3. *Введение в TypeScript – [Электронный ресурс]. Режим доступа: https://metanit.com/web/typescript/1.1.php*

4. *How to build a React library using TypeScript – [Электронный ресурс]. Режим доступа: https://prateeksurana.me/blog/react-library-with-typescript/*

UDC 621.3.049.77–048.24:537.2

BUILDING A REACT LIBRARY USING TYPESCRIPT

Voronko T.M.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Piskun G.A. – PhD, assistant professor, associate professor of the department of ICSD

Annotation. Nowadays React appears to be one of the most popular JavaScript libraries in the developing community. It has wide range of functionality, which makes accomplishing of different tasks much easier and faster, as well as being open-source, which allows to expand or modify this functionality. Procedure was developed for creating a React library using TypeScript.

Keywords:. JavaScript library, React, TypeScript.