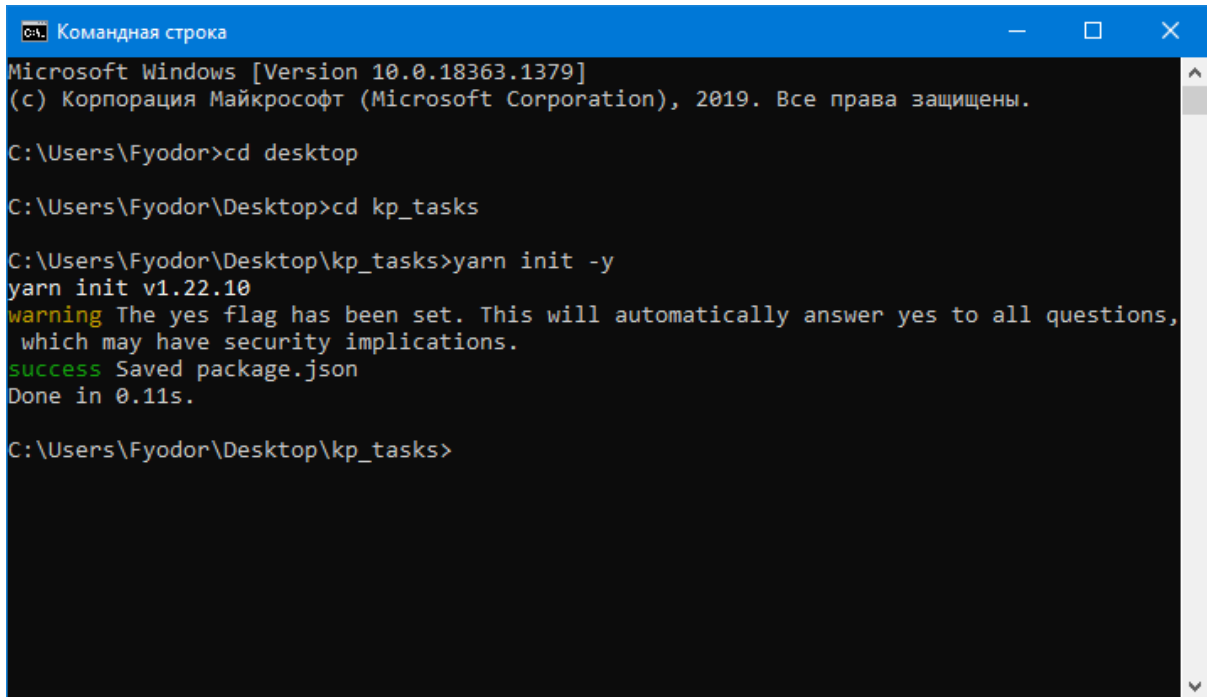


## Транспиляция. Babel. Компонентность

**Цель:** освоить основные операции по транспиляции кода с помощью Babel.

### I. Развёртывание проекта на JavaScript, включающего модули

#### 1. Создание проекта



```
Командная строка
Microsoft Windows [Version 10.0.18363.1379]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

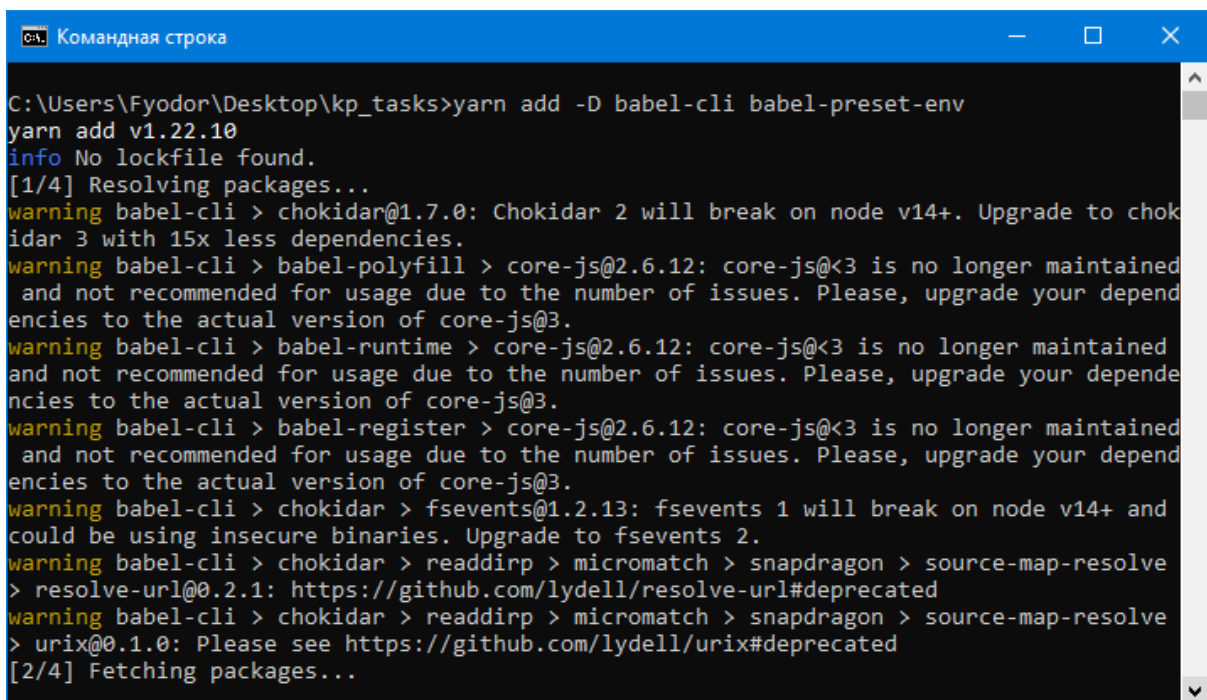
C:\Users\Fyodor>cd desktop

C:\Users\Fyodor\Desktop>cd kp_tasks

C:\Users\Fyodor\Desktop\kp_tasks>yarn init -y
yarn init v1.22.10
warning The yes flag has been set. This will automatically answer yes to all questions,
which may have security implications.
success Saved package.json
Done in 0.11s.

C:\Users\Fyodor\Desktop\kp_tasks>
```

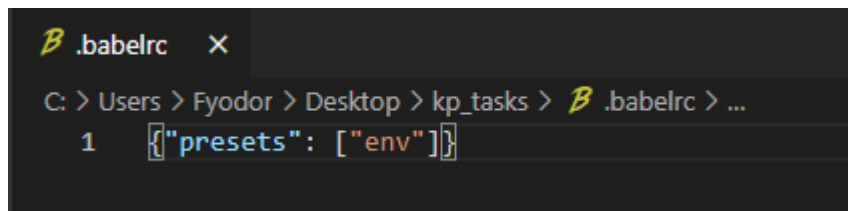
#### 2. Добавление зависимостей



```
Командная строка

C:\Users\Fyodor\Desktop\kp_tasks>yarn add -D babel-cli babel-preset-env
yarn add v1.22.10
info No lockfile found.
[1/4] Resolving packages...
warning babel-cli > chokidar@1.7.0: Chokidar 2 will break on node v14+. Upgrade to chokidar 3 with 15x less dependencies.
warning babel-cli > babel-polyfill > core-js@2.6.12: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning babel-cli > babel-runtime > core-js@2.6.12: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning babel-cli > babel-register > core-js@2.6.12: core-js@<3 is no longer maintained and not recommended for usage due to the number of issues. Please, upgrade your dependencies to the actual version of core-js@3.
warning babel-cli > chokidar > fsevents@1.2.13: fsevents 1 will break on node v14+ and could be using insecure binaries. Upgrade to fsevents 2.
warning babel-cli > chokidar > readdirp > micromatch > snapdragon > source-map-resolve > resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
warning babel-cli > chokidar > readdirp > micromatch > snapdragon > source-map-resolve > urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
[2/4] Fetching packages...
```

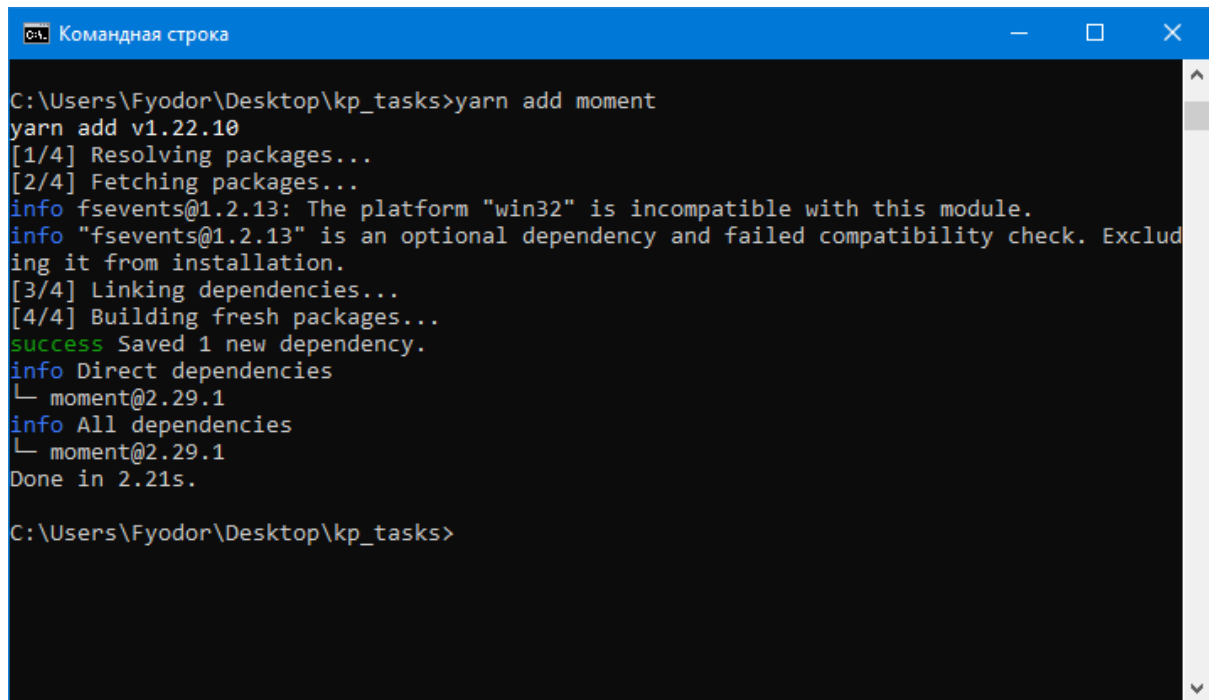
### 3. Настройка Babel



The screenshot shows a VS Code editor window with a single file named `.babelrc`. The file content is a JSON object with a `presets` array containing the string `"env"`. The path in the breadcrumb is `C:\Users\Fyodor\Desktop\kp_tasks > .babelrc > ...`.

```
.babelrc X
C: > Users > Fyodor > Desktop > kp_tasks > .babelrc > ...
1 [{"presets": ["env"]}]
```

### 4. Добавление библиотеки moment

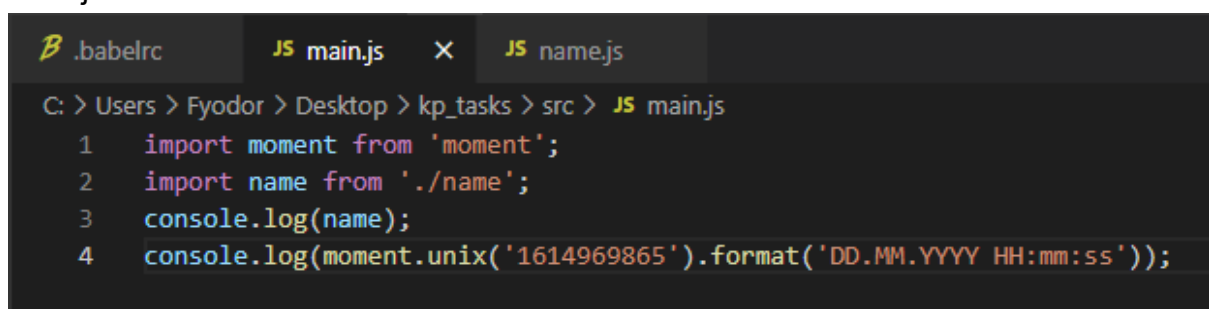


The screenshot shows a Windows Command Prompt window titled "Командная строка". The user has run the command `yarn add moment`. The output shows the installation progress, including resolving and fetching packages, and a success message indicating that 1 new dependency was saved. The final output shows the installed dependency `moment@2.29.1`.

```
Командная строка
C:\Users\Fyodor\Desktop\kp_tasks>yarn add moment
yarn add v1.22.10
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@1.2.13: The platform "win32" is incompatible with this module.
info "fsevents@1.2.13" is an optional dependency and failed compatibility check. Exclud
ing it from installation.
[3/4] Linking dependencies...
[4/4] Building fresh packages...
success Saved 1 new dependency.
info Direct dependencies
└─ moment@2.29.1
info All dependencies
└─ moment@2.29.1
Done in 2.21s.

C:\Users\Fyodor\Desktop\kp_tasks>
```

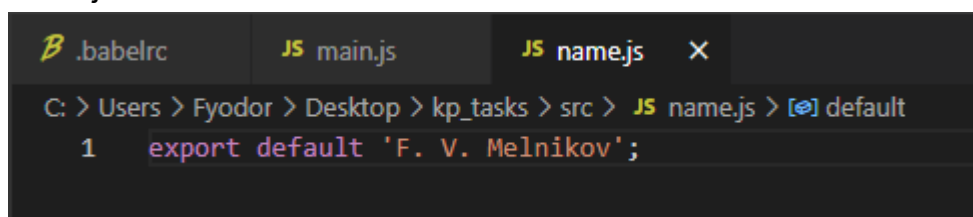
### 5. Создание файлов, находящихся в отношении нативной модульности main.js



The screenshot shows a VS Code editor window with three files: `.babelrc`, `main.js`, and `name.js`. The `main.js` file is selected and shows the following code:

```
.babelrc JS main.js X JS name.js
C: > Users > Fyodor > Desktop > kp_tasks > src > JS main.js
1 import moment from 'moment';
2 import name from './name';
3 console.log(name);
4 console.log(moment.unix('1614969865').format('DD.MM.YYYY HH:mm:ss'));
```

### name.js

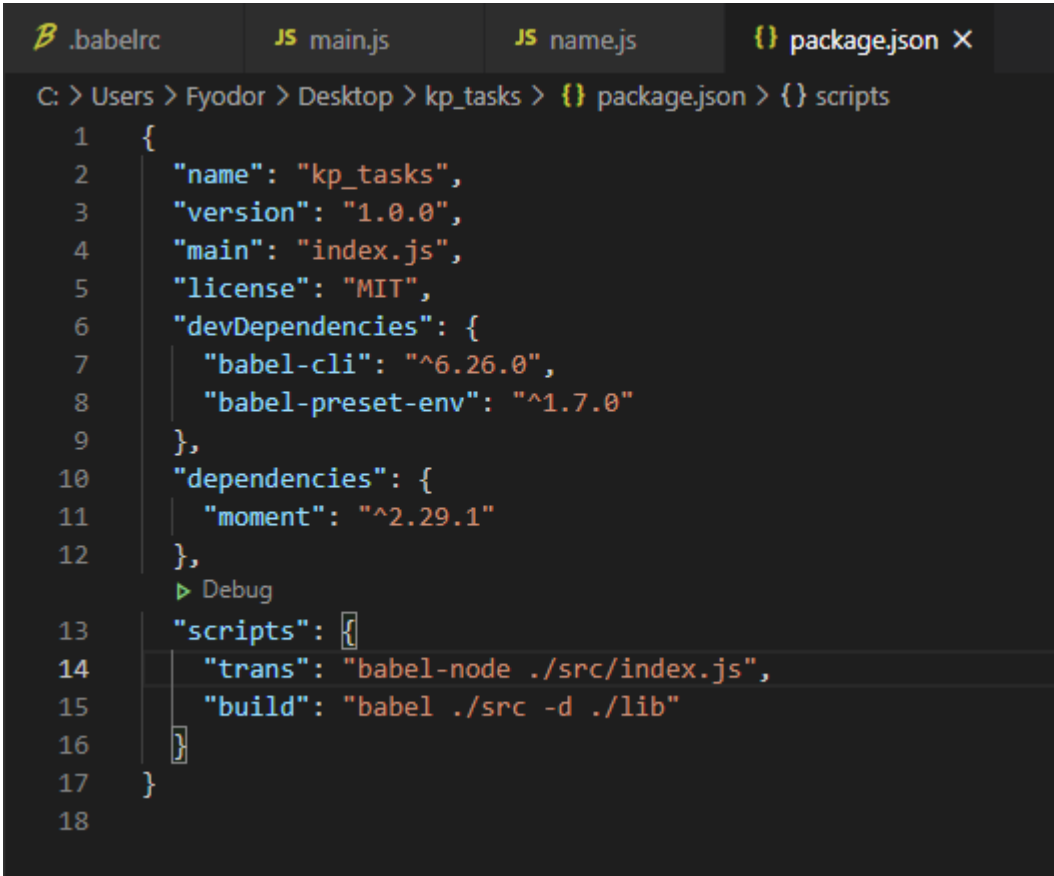


The screenshot shows a VS Code editor window with three files: `.babelrc`, `main.js`, and `name.js`. The `name.js` file is selected and shows the following code:

```
.babelrc JS main.js JS name.js X
C: > Users > Fyodor > Desktop > kp_tasks > src > JS name.js > default
1 export default 'F. V. Melnikov';
```

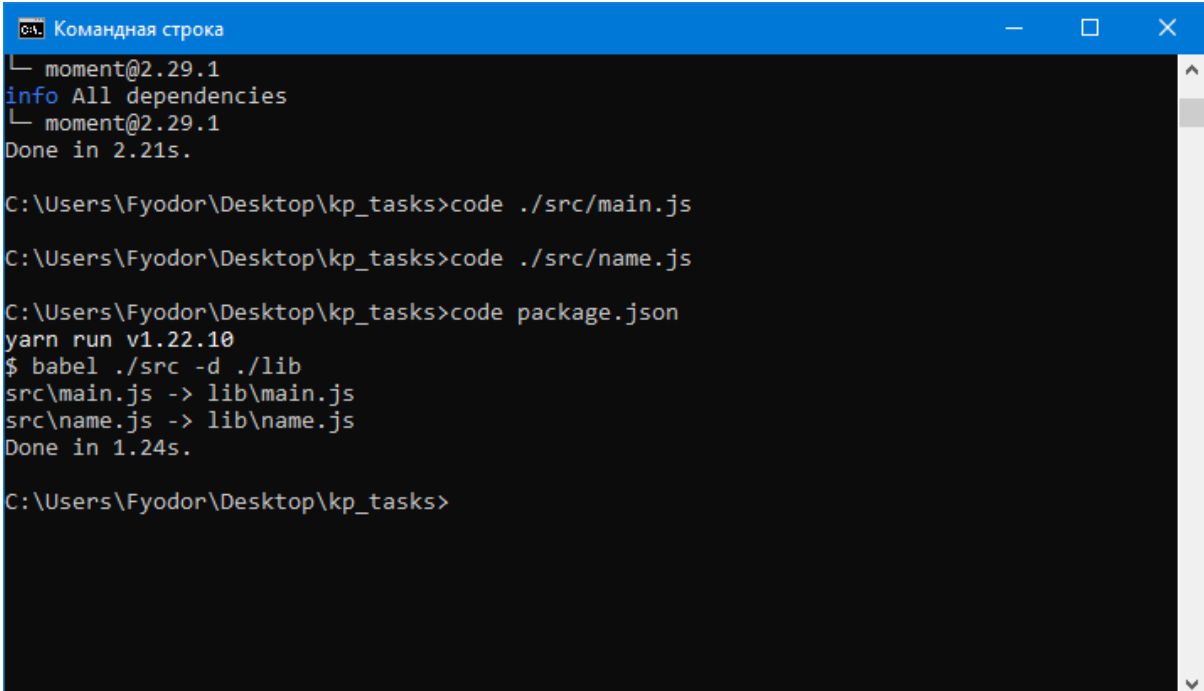
## 6. Выполнение команды транспилиции

### Добавление сценария в package.json

A screenshot of the Visual Studio Code editor interface. The top bar shows four tabs: ".babelrc", "JS main.js", "JS name.js", and "package.json" (which is active and has a close button). The main editor area displays the content of "package.json" with line numbers 1 through 18. The JSON structure includes "name", "version", "main", "license", "devDependencies" (with "babel-cli" and "babel-preset-env"), "dependencies" (with "moment"), and a "scripts" object containing "trans" and "build" commands. The "scripts" object is currently selected with a mouse cursor.

```
1  {
2    "name": "kp_tasks",
3    "version": "1.0.0",
4    "main": "index.js",
5    "license": "MIT",
6    "devDependencies": {
7      "babel-cli": "^6.26.0",
8      "babel-preset-env": "^1.7.0"
9    },
10   "dependencies": {
11     "moment": "^2.29.1"
12   },
13   "scripts": {
14     "trans": "babel-node ./src/index.js",
15     "build": "babel ./src -d ./lib"
16   }
17 }
18
```

### Сборка проекта:

A screenshot of a Windows Command Prompt window titled "Командная строка". It shows the output of a yarn command and subsequent manual commands. The output includes dependency information for moment@2.29.1 and the execution of the "trans" and "build" scripts from the package.json file, which use babel to transpile source files into a lib directory.

```
moment@2.29.1
info All dependencies
└─ moment@2.29.1
Done in 2.21s.

C:\Users\Fyodor\Desktop\kp_tasks>code ./src/main.js

C:\Users\Fyodor\Desktop\kp_tasks>code ./src/name.js

C:\Users\Fyodor\Desktop\kp_tasks>code package.json
yarn run v1.22.10
$ babel ./src -d ./lib
src\main.js -> lib\main.js
src\name.js -> lib\name.js
Done in 1.24s.

C:\Users\Fyodor\Desktop\kp_tasks>
```

## 7. Получены транспилированные файлы

```
Командная строка
C:\Users\Fyodor\Desktop\kp_tasks>code package.json
yarn run v1.22.10
$ babel ./src -d ./lib
src\main.js -> lib\main.js
src\name.js -> lib\name.js
Done in 1.24s.

C:\Users\Fyodor\Desktop\kp_tasks>cd lib

C:\Users\Fyodor\Desktop\kp_tasks\lib>dir
Том в устройстве C не имеет метки.
Серийный номер тома: 0088-CC0A

Содержимое папки C:\Users\Fyodor\Desktop\kp_tasks\lib

06.03.2021  08:49    <DIR>          .
06.03.2021  08:49    <DIR>          ..
06.03.2021  08:49                380 main.js
06.03.2021  08:49                115 name.js
                2 файлов          495 байт
                2 папок    9 241 010 176 байт свободно

C:\Users\Fyodor\Desktop\kp_tasks\lib>
```

Вызов main.js не вызывает ошибок:

```
Командная строка
C:\Users\Fyodor\Desktop\kp_tasks\lib>dir
Том в устройстве C не имеет метки.
Серийный номер тома: 0088-CC0A

Содержимое папки C:\Users\Fyodor\Desktop\kp_tasks\lib

06.03.2021  08:49    <DIR>          .
06.03.2021  08:49    <DIR>          ..
06.03.2021  08:49                380 main.js
06.03.2021  08:49                115 name.js
                2 файлов          495 байт
                2 папок    9 241 010 176 байт свободно

C:\Users\Fyodor\Desktop\kp_tasks\lib>code main.js
C:\Users\Fyodor\Desktop\kp_tasks\lib>code name.js
C:\Users\Fyodor\Desktop\kp_tasks\lib>node main.js
F. V. Melnikov
05.03.2021  21:44:25

C:\Users\Fyodor\Desktop\kp_tasks\lib>
```

## 8. Содержимое файлов

main.js

```
'use strict';

var _moment = require('moment');

var _moment2 = _interopRequireDefault(_moment);

var _name = require('./name');

var _name2 = _interopRequireDefault(_name);

function _interopRequireDefault(obj) { return obj && obj.__esModule ? obj : { default: obj }; }

console.log(_name2.default);
console.log(_moment2.default.unix('1614969865').format('DD.MM.YYYY HH:mm:ss'));
```

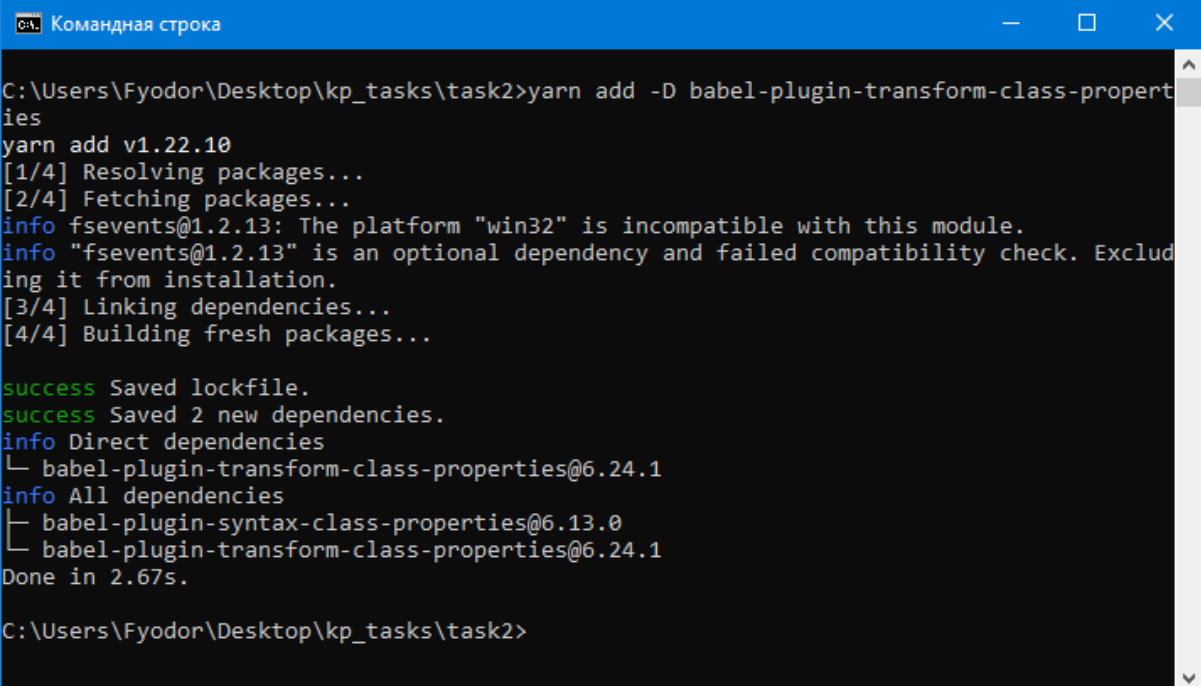
name.js

```
'use strict';

Object.defineProperty(exports, "__esModule", {
  value: true
});
exports.default = 'F. V. Melnikov';
```

## II. Преобразование кода на JavaScript, содержащего инструкции из следующих версий стандарта ECMAScript

### 1. Добавление поддержки class properties



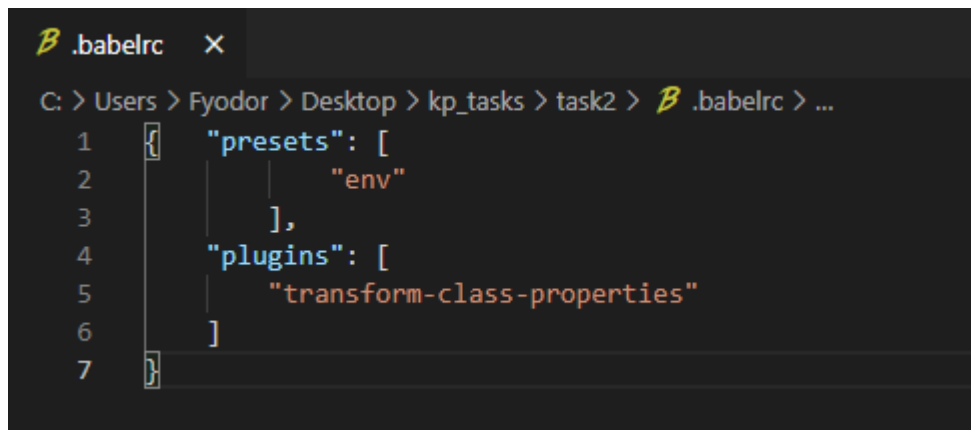
```
cmd. Командная строка

C:\Users\Fyodor\Desktop\kp_tasks\task2>yarn add -D babel-plugin-transform-class-properties
yarn add v1.22.10
[1/4] Resolving packages...
[2/4] Fetching packages...
info fsevents@1.2.13: The platform "win32" is incompatible with this module.
info "fsevents@1.2.13" is an optional dependency and failed compatibility check. Excluding it from installation.
[3/4] Linking dependencies...
[4/4] Building fresh packages...

success Saved lockfile.
success Saved 2 new dependencies.
info Direct dependencies
└─ babel-plugin-transform-class-properties@6.24.1
info All dependencies
└─ babel-plugin-syntax-class-properties@6.13.0
└─ babel-plugin-transform-class-properties@6.24.1
Done in 2.67s.

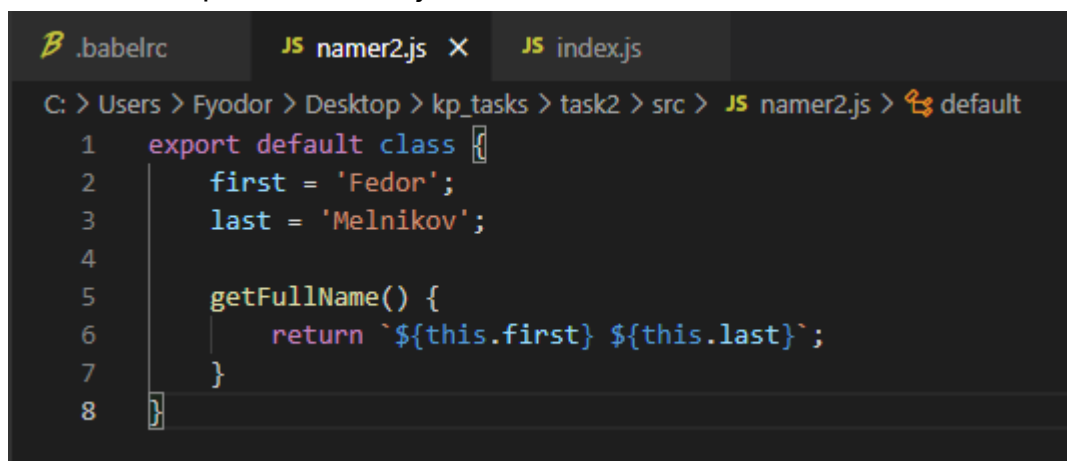
C:\Users\Fyodor\Desktop\kp_tasks\task2>
```

## 2. Изменение файла .babelrc



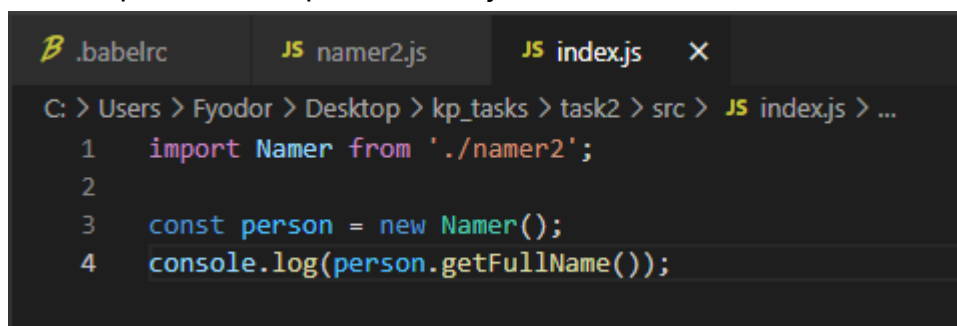
```
.babelrc X
C: > Users > Fyodor > Desktop > kp_tasks > task2 > .babelrc > ...
1  {
2    "presets": [
3      "env"
4    ],
5    "plugins": [
6      "transform-class-properties"
7    ]
8  }
```

## 3. Создание файла namer2.js



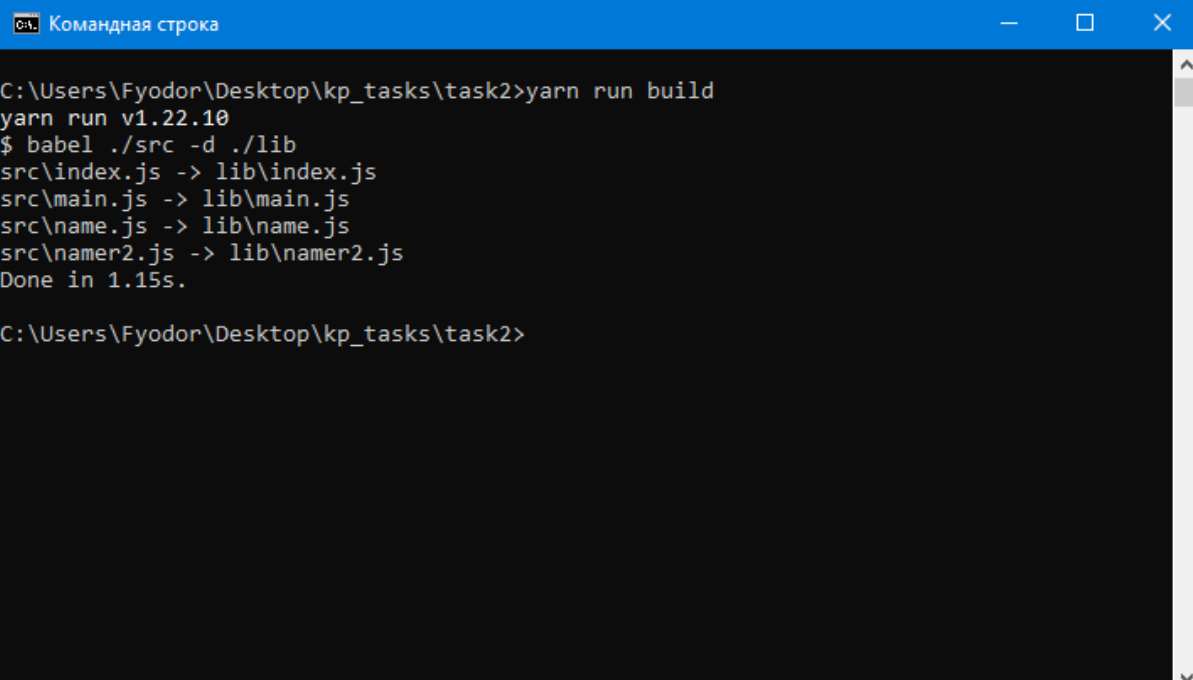
```
.babelrc JS namer2.js X JS index.js
C: > Users > Fyodor > Desktop > kp_tasks > task2 > src > JS namer2.js > default
1  export default class {
2    first = 'Fedor';
3    last = 'Melnikov';
4
5    getFullName() {
6      return `${this.first} ${this.last}`;
7    }
8  }
```

## 4. Импорт класса в файле index.js



```
.babelrc JS namer2.js JS index.js X
C: > Users > Fyodor > Desktop > kp_tasks > task2 > src > JS index.js > ...
1  import Namer from './namer2';
2
3  const person = new Namer();
4  console.log(person.getFullName());
```

## 5. Сборка проекта

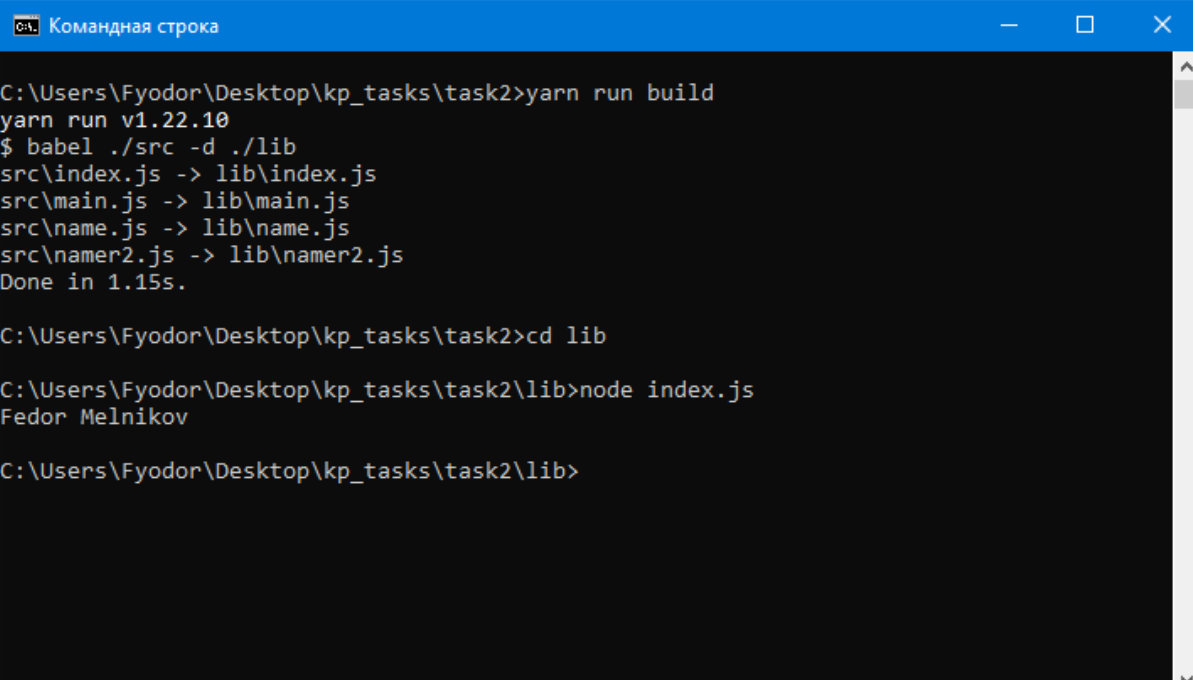


```
Командная строка

C:\Users\Fyodor\Desktop\kp_tasks\task2>yarn run build
yarn run v1.22.10
$ babel ./src -d ./lib
src\index.js -> lib\index.js
src\main.js -> lib\main.js
src\name.js -> lib\name.js
src\namer2.js -> lib\namer2.js
Done in 1.15s.

C:\Users\Fyodor\Desktop\kp_tasks\task2>
```

## 6. Выполнение файла index.js



```
Командная строка

C:\Users\Fyodor\Desktop\kp_tasks\task2>yarn run build
yarn run v1.22.10
$ babel ./src -d ./lib
src\index.js -> lib\index.js
src\main.js -> lib\main.js
src\name.js -> lib\name.js
src\namer2.js -> lib\namer2.js
Done in 1.15s.

C:\Users\Fyodor\Desktop\kp_tasks\task2>cd lib

C:\Users\Fyodor\Desktop\kp_tasks\task2\lib>node index.js
Fedor Melnikov

C:\Users\Fyodor\Desktop\kp_tasks\task2\lib>
```

**Вывод:** в ходе работы были освоены основные операции по транспилиции кода с помощью Babel.