

GIT

Начнём с создания репозитория

Изменяем главную ветку

- По-умолчанию основная ветка main, если ещё не создали репозиторий, то можно без проблем изменить ее в настройках гита. Меняем на master и нажимаем update.

The screenshot shows the GitHub interface for a user named 'JExample'. The page title is 'Repository default branch'. The main content area contains a text block explaining that the default branch for new repositories can be chosen, with 'master' as the current selection. Below this, there is a text input field containing 'master', a green checkmark icon, and an 'Update' button. A green notification box below the input field states 'Your default branch name will be master'. On the left side, there is a sidebar menu with various account settings, and the 'Repositories' option is highlighted. At the bottom, there is a section titled 'Repositories' showing a list of repositories, including 'JExample/hw3' with 4 KB size and 0 collaborators.

Repositories

https://github.com/settings/repositories

Search or jump to...

Pulls Issues Marketplace Explore

JExample
Your personal account

Go to your personal profile

Account settings

Profile

Account

Appearance New

Account security

Billing & plans

Security log

Security & analysis

Emails

Notifications

SSH and GPG keys

Repositories

Packages

Repository default branch

Choose the default branch for your new personal repositories. You might want to change the default name due to different workflows, or because your integrations still require "master" as the default branch name. You can always change the default branch name on individual repositories. [Learn more about default branches.](#)

master Update

Your default branch name will be master

Repositories

Repositories Deleted repositories

JExample

JExample/hw3 4 KB 0 collaborators

Создаём репозиторий

- Если уже создан, не беда. Нет требования, чтобы основная ветка называлась master, а не main.
- В будущем можно будет изменить ветку, просто это не так удобно как сделать это изначально.
- 1) Название – что-нибудь связанное с курсами (не обязательно, главное приличное)
- 2) Репозиторий private
- 3) добавляем по желанию README файл, он обычно содержит описание и в гите это описание отображается под структурой (далее будет видно о чем я)
- 4) Добавляем .gitignore для Java (.gitignore template: Java)
- 5) Снизу видим как будет называться ветка по умолчанию и ссылка на настройки, где можно ее изменить (предыдущий слайд)

6) Создаём

Create a New Repository x +

← → ↻ https://github.com/new

Search or jump to... / Pulls Issues Marketplace Explore

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner * JExample / Repository name * JavaEE ✓

Great repository names are short and memorable. Need inspiration? How about [fantastic-broccoli?](#)

Description (optional)
Проект для курсов JavaEE

Public
Anyone on the internet can see this repository. You choose who can commit.

Private
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

Add a README file
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore
Choose which files not to track from a list of templates. [Learn more.](#)
.gitignore template: Java

Choose a license
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set `master` as the default branch. Change the default name in your [settings](#).

Create repository

Любуемся результатом

Содержимое файла
README.md отображается
под структурой репозитория.

The screenshot shows a GitHub repository page for 'JExample/JavaEE'. The repository is private and has 1 pull request, 0 stars, and 0 forks. The 'Code' tab is selected, showing the file structure with files like '.gitignore', 'README.md', and 'README.md'. The 'README.md' file is expanded, showing the title 'JavaEE' and the subtitle 'Проект для курсов JavaEE'. The footer contains copyright information for 2021 GitHub, Inc. and various links like Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

JExample/JavaEE: Проект для курсов JavaEE

https://github.com/JExample/JavaEE

Search or jump to... Pulls Issues Marketplace Explore

JExample / JavaEE Private Unwatch 1 Star 0 Fork 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

master Go to file Add file Code

JExample Initial commit ... now 1

.gitignore Initial commit now

README.md Initial commit now

README.md

JavaEE

Проект для курсов JavaEE

About
Проект для курсов JavaEE
Readme

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

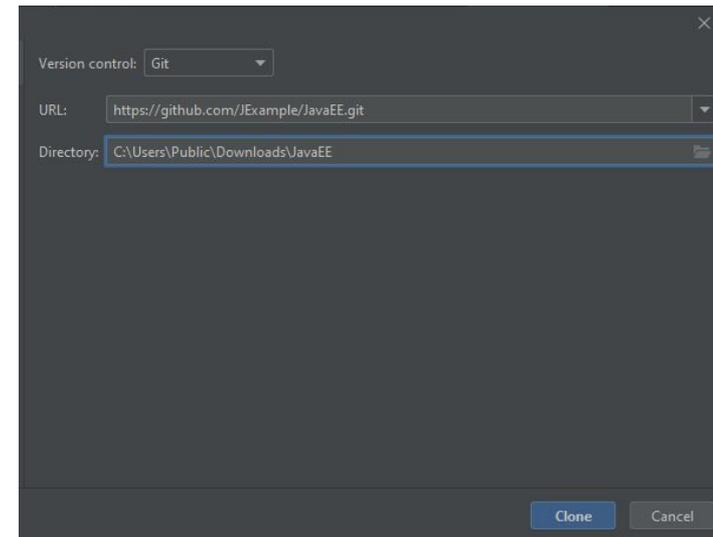
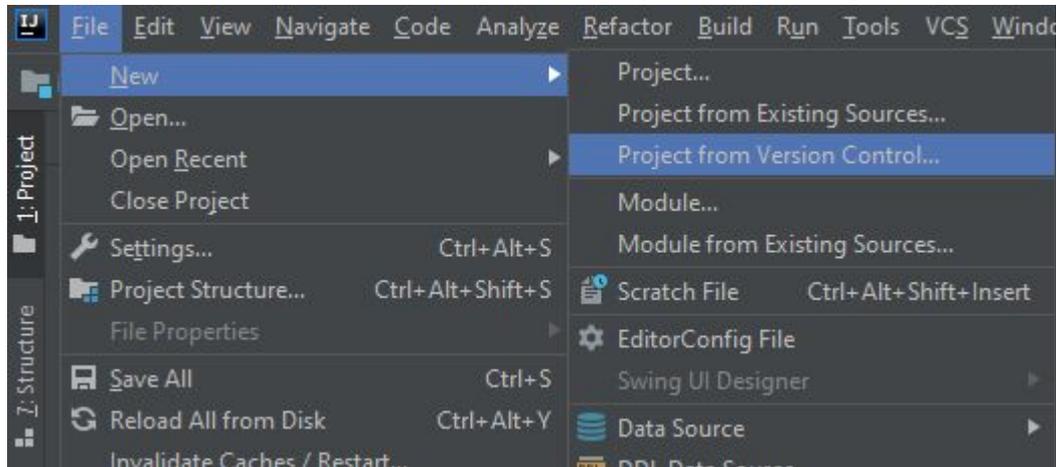
© 2021 GitHub, Inc. Terms Privacy Security Status Docs
Contact GitHub Pricing API Training Blog About

Нужно теперь забрать себе проект

- самый простой способ – сделать это с помощью IDEA

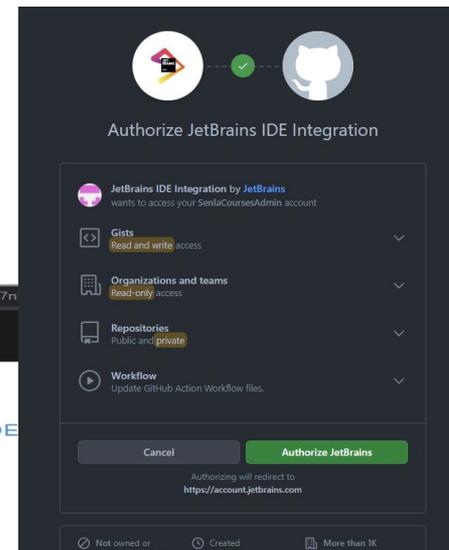
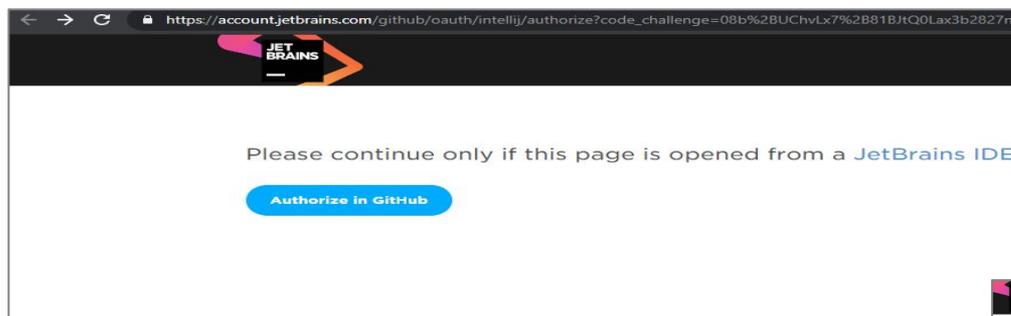
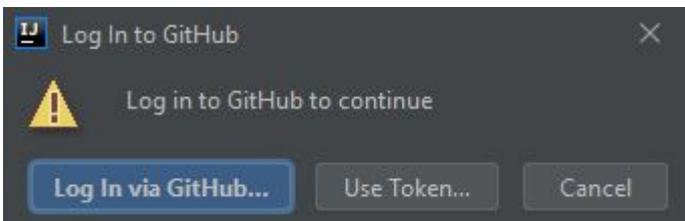
- File → New → Project from Version Control...

- Вводим ссылку, выбираем расположение и нажимаем Clone



Log In to GitHub to continue...

- 1) Дальше если в идее ещё не авторизованы, то нужно пройти аутентификацию. Нажимаем Log In via Github... →
- 2) Нажимаем Authorize in GitHub в открывшемся браузере →
- 3) Нажимаем Authorize JetBrains (либо просто появится сообщение об успешной авторизации)

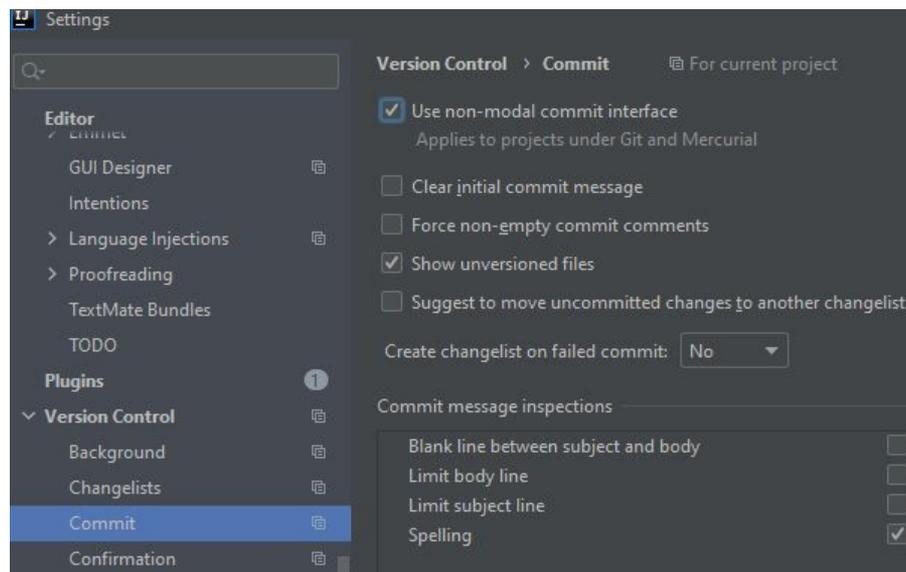
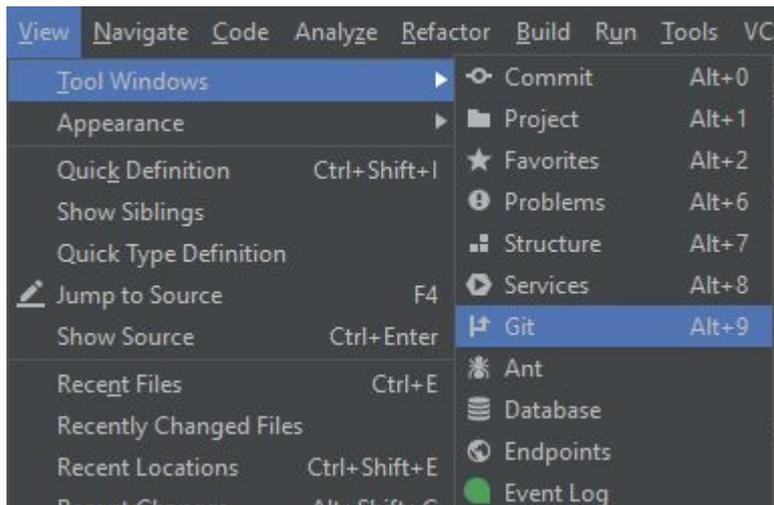


На примере VCS (git) в IDEA

1) Если нет вкладки, ее можно добавить из View → Tool Windows → Git (в старых версиях VCS)

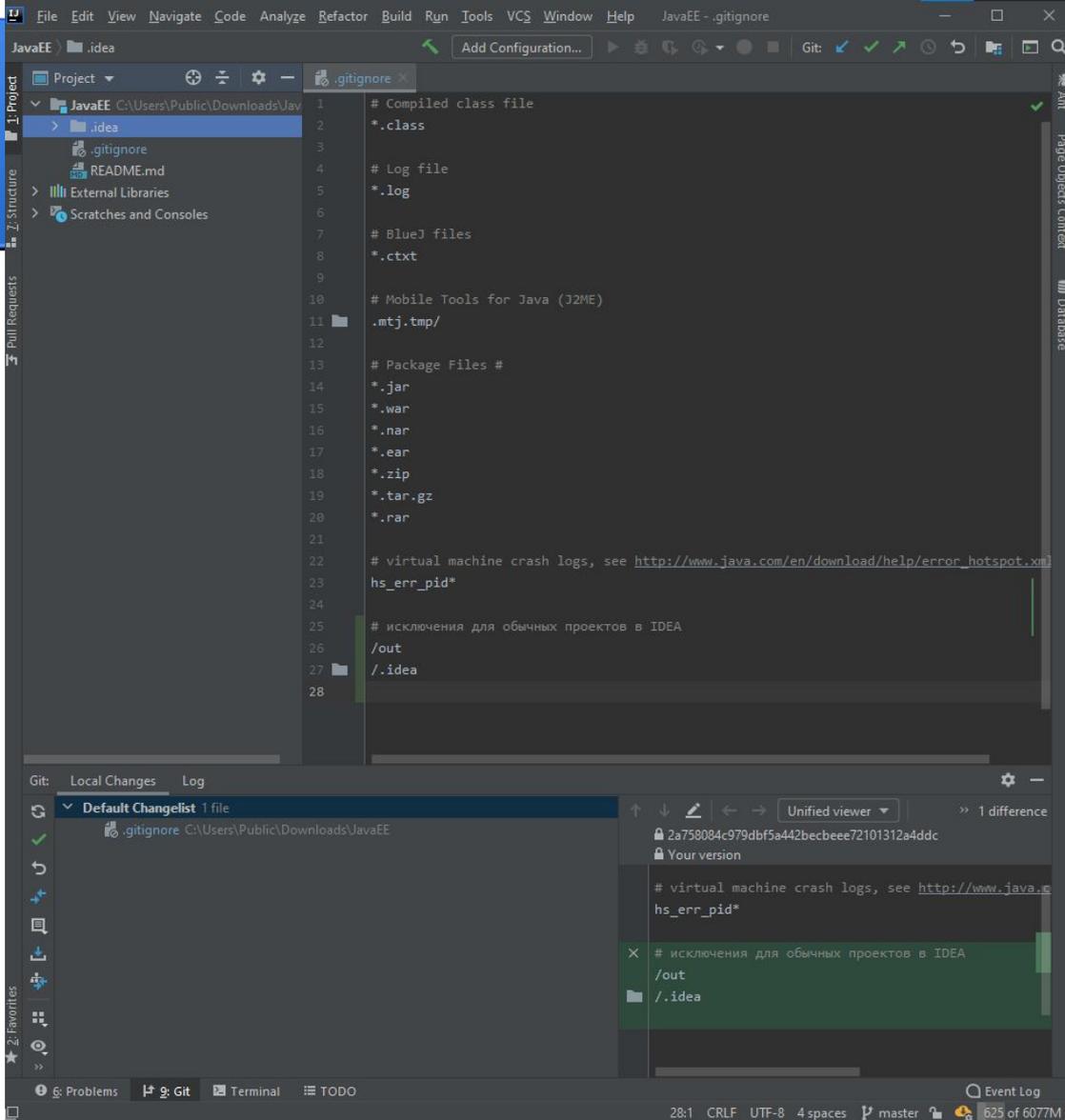


Я не люблю новый режим коммита, он связан с отображением изменений, поэтому если включен режим non-modal , и сразу его выключаю



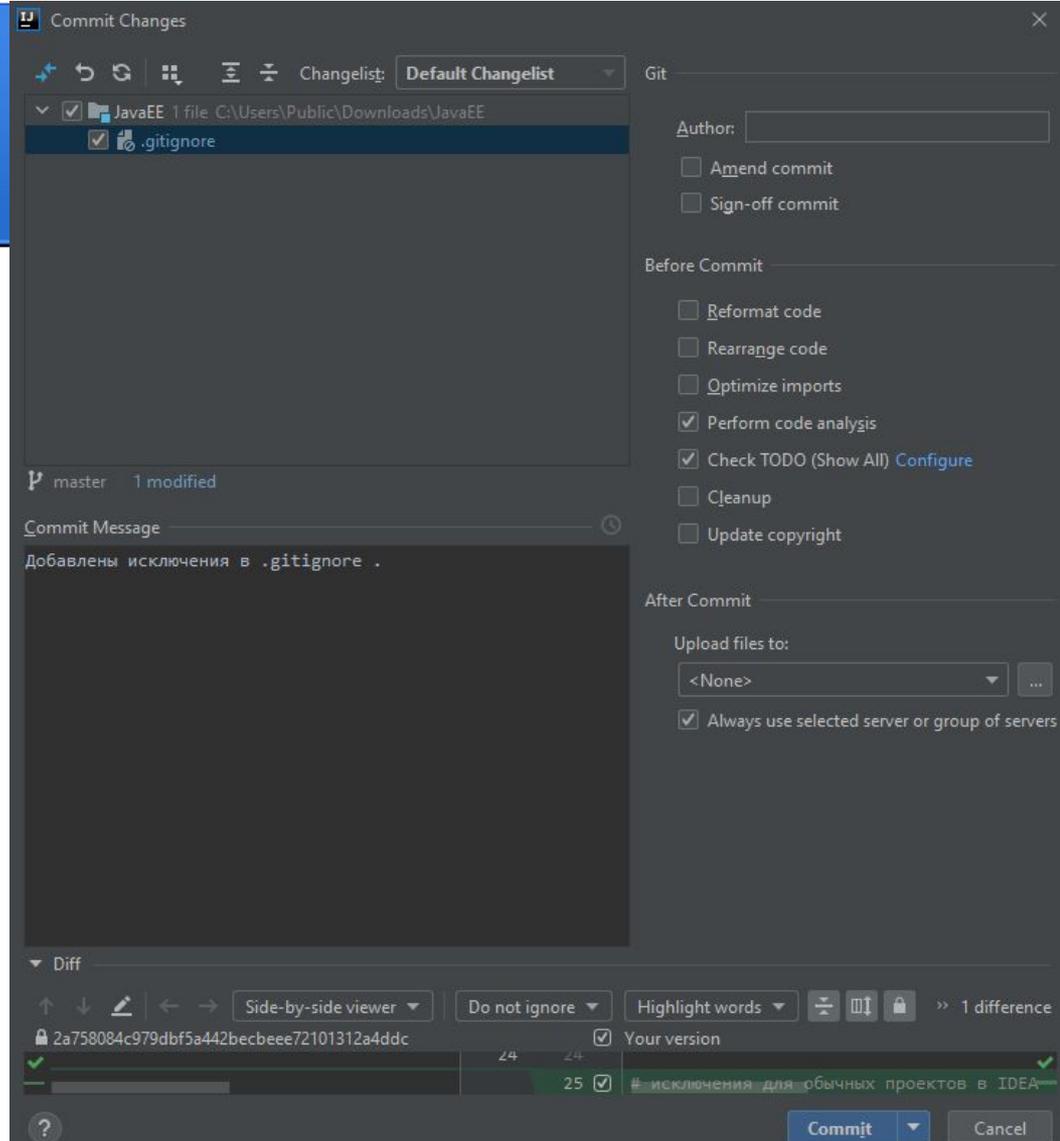
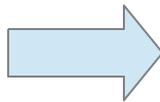
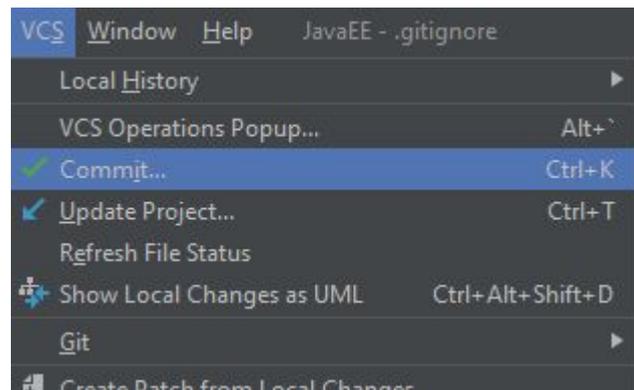
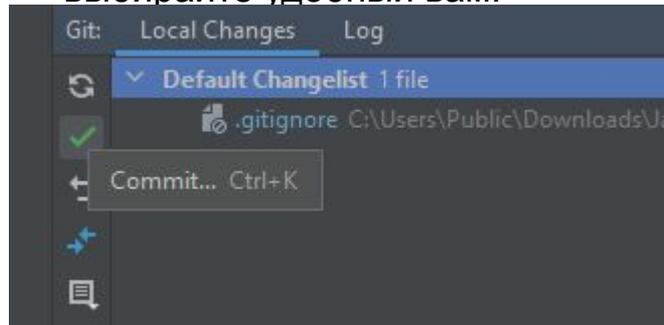
Продолжение

- Добавляем в файл .gitignore /out и /.idea для исключения файлов идеи и результатов сборки проекта.
- Теперь во вкладке Git мы видим раздел Local Changes – это все измененные файлы, которые наблюдает git.



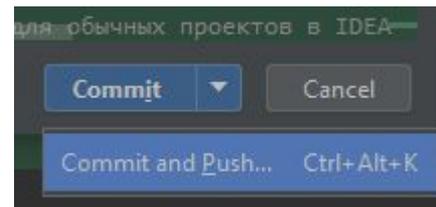
Теперь в основную ветку добавим изменения .gitignore

- Создадим новый коммит и оправим (push) его в ветку master (main). Способов сделать коммит много, выбирайте удобный вам.



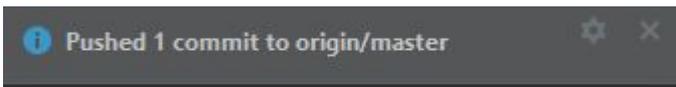
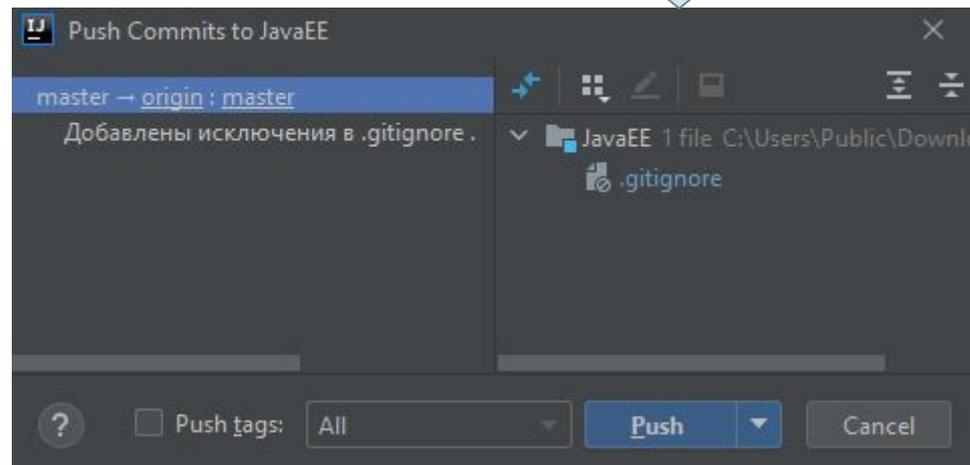
Заполняем данные для Commit

После заполнения текста, убеждаемся что необходимые файлы выбраны, и нажимаем Commit либо Commit and Push. Разница только в том, что в первом случае для выполнения Push нужны дополнительные действия (VCS → Push).



В диалоговом окне Push видно из какой локальной ветки будут отправлены изменения (master) и в какую удалённую (origin:master). Далее будет список коммитов, в нашем случае он только 1.

Нажимаем push и идея всю делает за нас.



Убеждаемся, что всё попало в git

master

Go to file Add file Code

vvoronovich and vvoronovich Добавлены исключения в .gitign... 4 minutes ago 2

.gitignore [Добавлены исключения в .gitignore .](#) 4 minutes ago

README.md Initial commit 1 hour ago

README.md

JavaEE

Проект для курсов JavaEE

master

Commits on May 26, 2021

Добавлены исключения в .gitignore . 226ffc8

vvoronovich authored and vvoronovich committed 5 minutes ago

Initial commit 2a75808 Verified

JExample committed 1 hour ago

Newer Older

Чтобы просмотреть список коммитов, нажимаем куда указывает стрелка. (если кто-то забыл)

Основная ветка готова, теперь можно переходить к выполнению заданий

- 1) Для 3 работы никакие изменения не должны быть в ветке master (main).
- 2) Для 3 работы предлагаю каждое задание делать в отдельной ветке. Для получения практики.
- 3) Для 4 все задания из листа должны будут быть выполнены в 1 (одной) ветке, ответвленной от master (main).

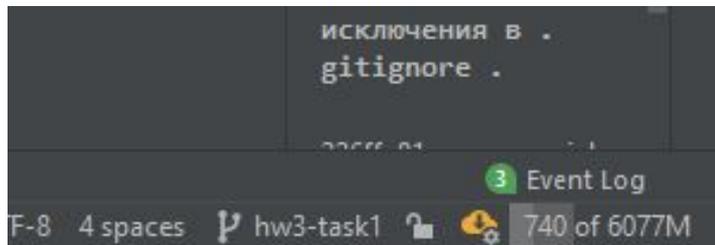
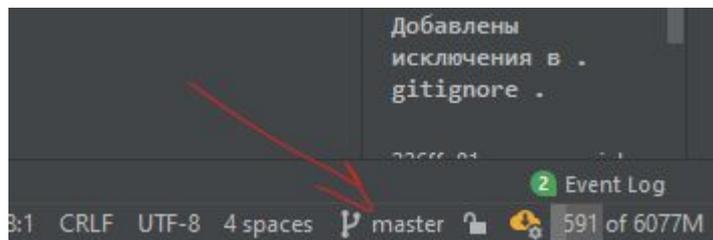
Выбрали New Branch...

Вводим название ветки, для 3 практики предлагаю следующее наименование `hw3-task*`, где вместо * номер задания (вариант не нужен).

Если флаг Checkout branch выбран, то мы попадем в нее сразу после создания



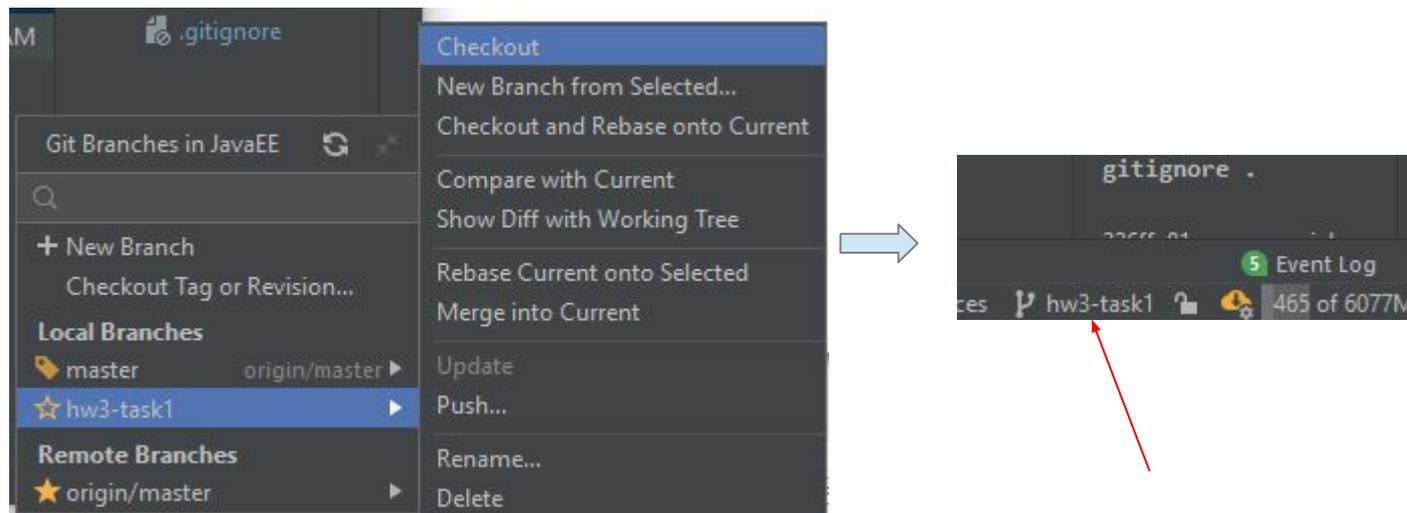
Идея покажет в какой ветке мы находимся, она должна измениться с `master` на `hw3-task1`



Checkout

- Checkout необходим для смены ветки, при этом файлы в локальном репозитории будут заменены необходимыми данными другой ветки. Файлы которых нет могут пропасть, новые появятся, но данные которые уже были коммитнуты (commit) никуда не пропадают, гит их хранит. IDEA при переходе между ветками если нет проблем, то просто переносит изменения, если есть проблемы – то предупреждает, в таком случае лучше сначала сделать commit и только потом менять ветку, чтобы не потерять данные.

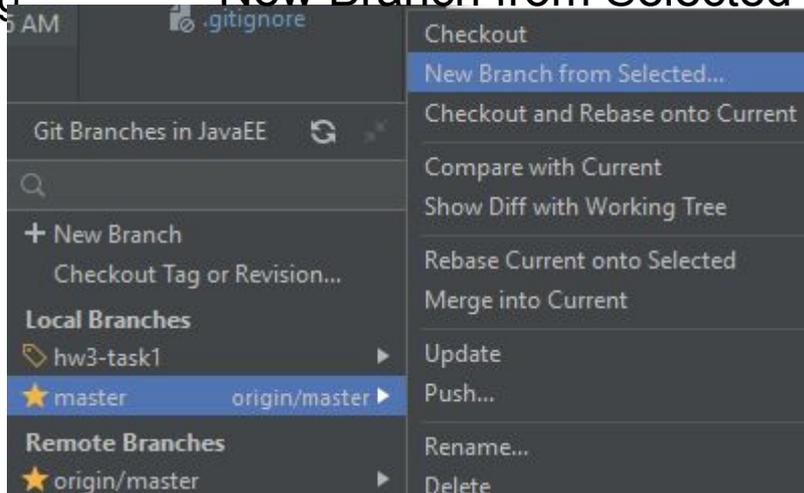
При нажатии на название ветки (в правом нижнем углу), появляется следующее окно с опциями, и мы можем сделать Checkout в другую ветку.



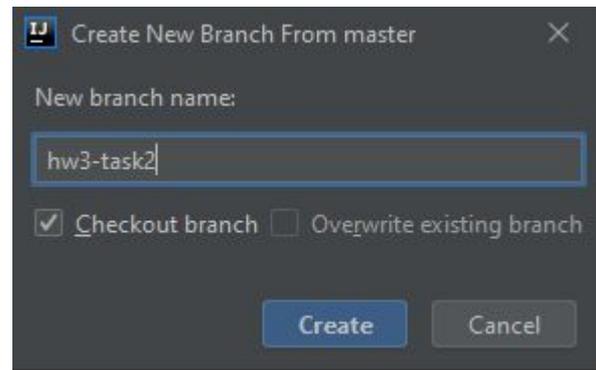
Ветвление от ветки

- Суть действия такая же как и ветвление от коммита, просто для этого не нужна вкладка Git → Log

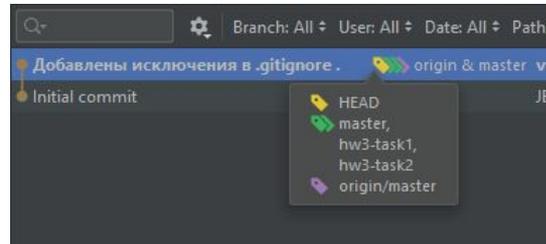
Для этого нужно в правом нижнем углу нажать на название ветки, выбрать ветку от которой будем создавать новую, **New Branch from Selected**



Аналогично ветвлению от коммита вводим название новой ветки и нажимаем создать

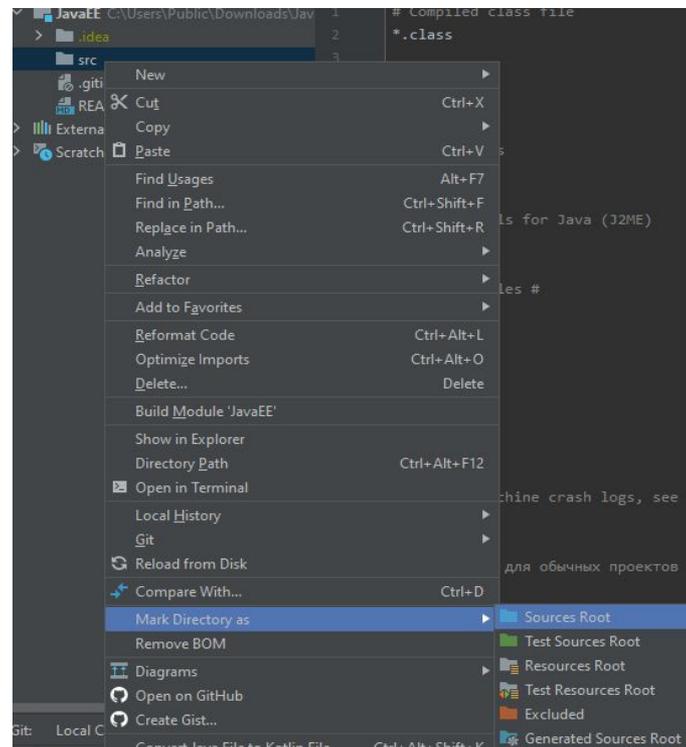
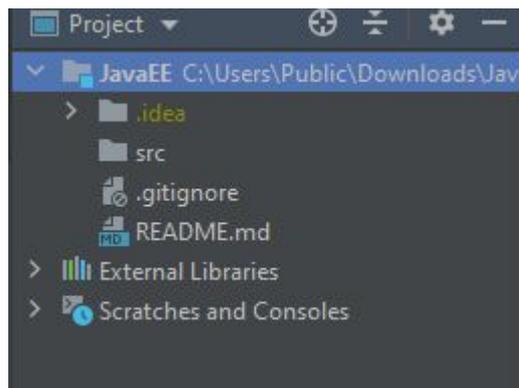
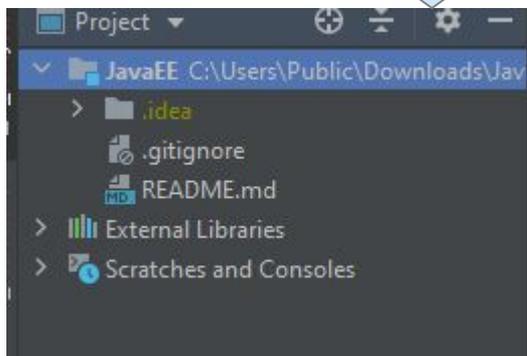
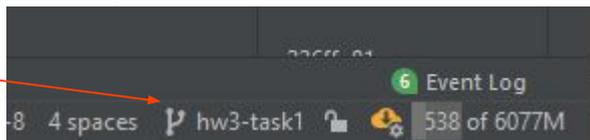
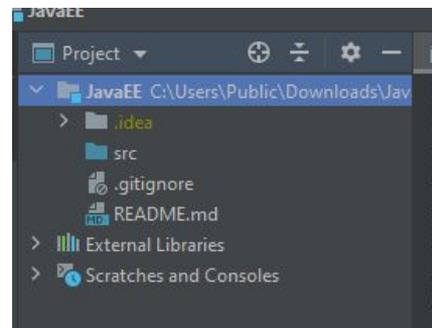


- Теперь в Log модно видеть один коммит является последним для веток master, hw3-task1, hw3-task2



Пример ветки 1 (задание 1)

- 1) Мы создали ветку (любым способом и перешли в нее [checkout])
- 2) Создадим необходимую нам папку для исходников. Src
- 3) если он не выделена цветом, то необходимо пометить ее как Source

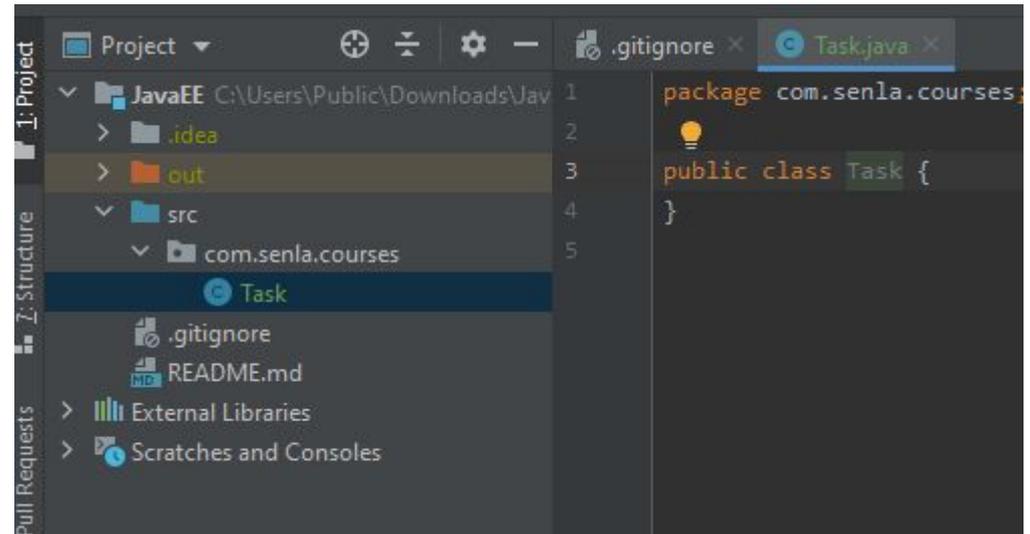
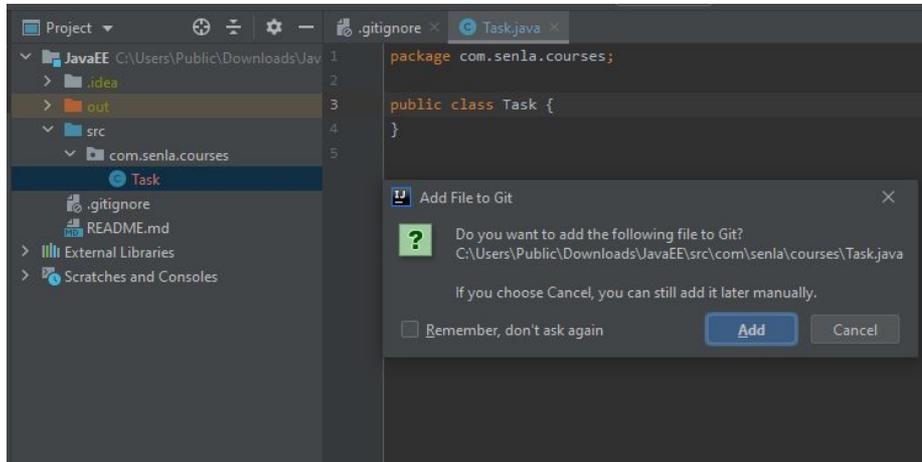


Создание новых файлов и добавление в git

1) Создаём необходимые пакеты и решаем поставленную

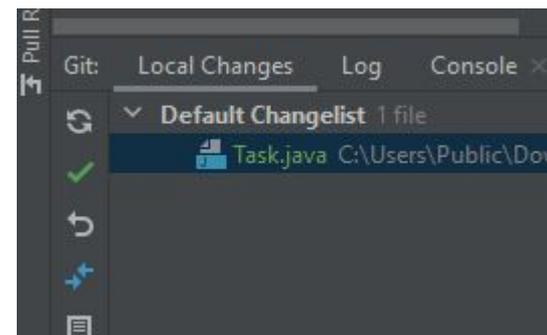
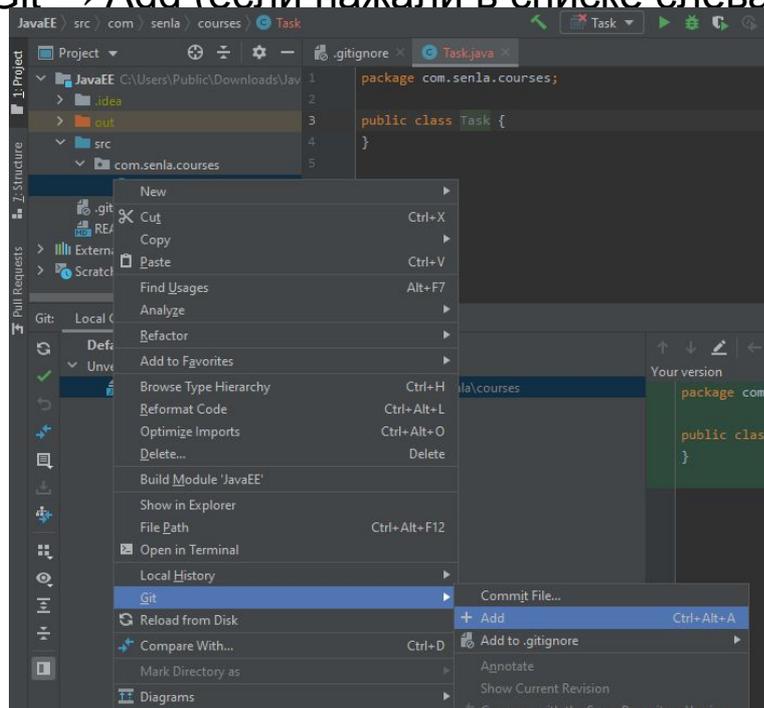
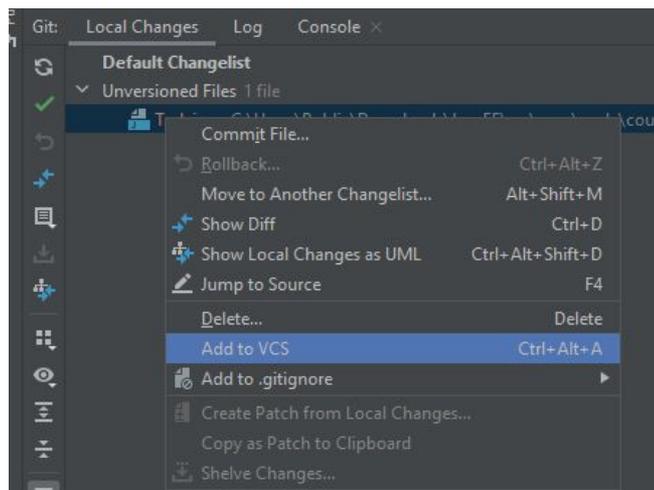
задачу

2) Когда создаём новый файл git спросит добавить ли его в отслеживаемые, если не добавить он будет красным и git не будет предлагать его коммитнуть. Если согласимся, он будет зелёным. (зеленый[новые файлы] и синий[отредактированные] - отслеживаемые)



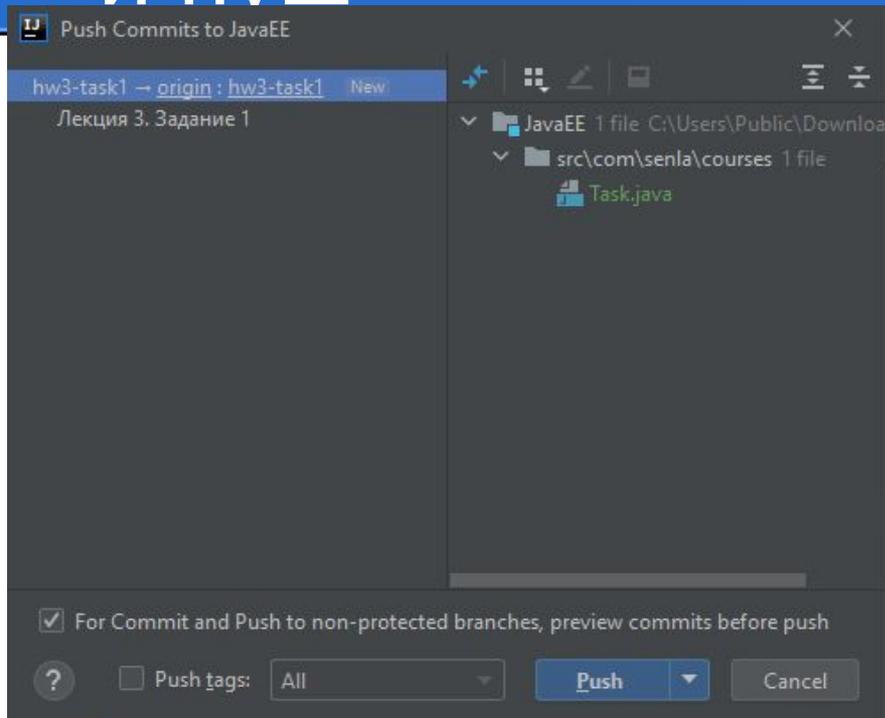
Ручное добавление файла

- Если случайно не добавили файл в git, либо специально, можно самостоятельно его добавить. Во вкладке Log он находится не в Default Changelist, а в Unversioned Files. Нажимаем ПКМ по файлу и выбираем Add to VCS (если нажали в списке Unversioned Files) либо ПКМ по файлу и Git → Add (если нажали в списке слева)



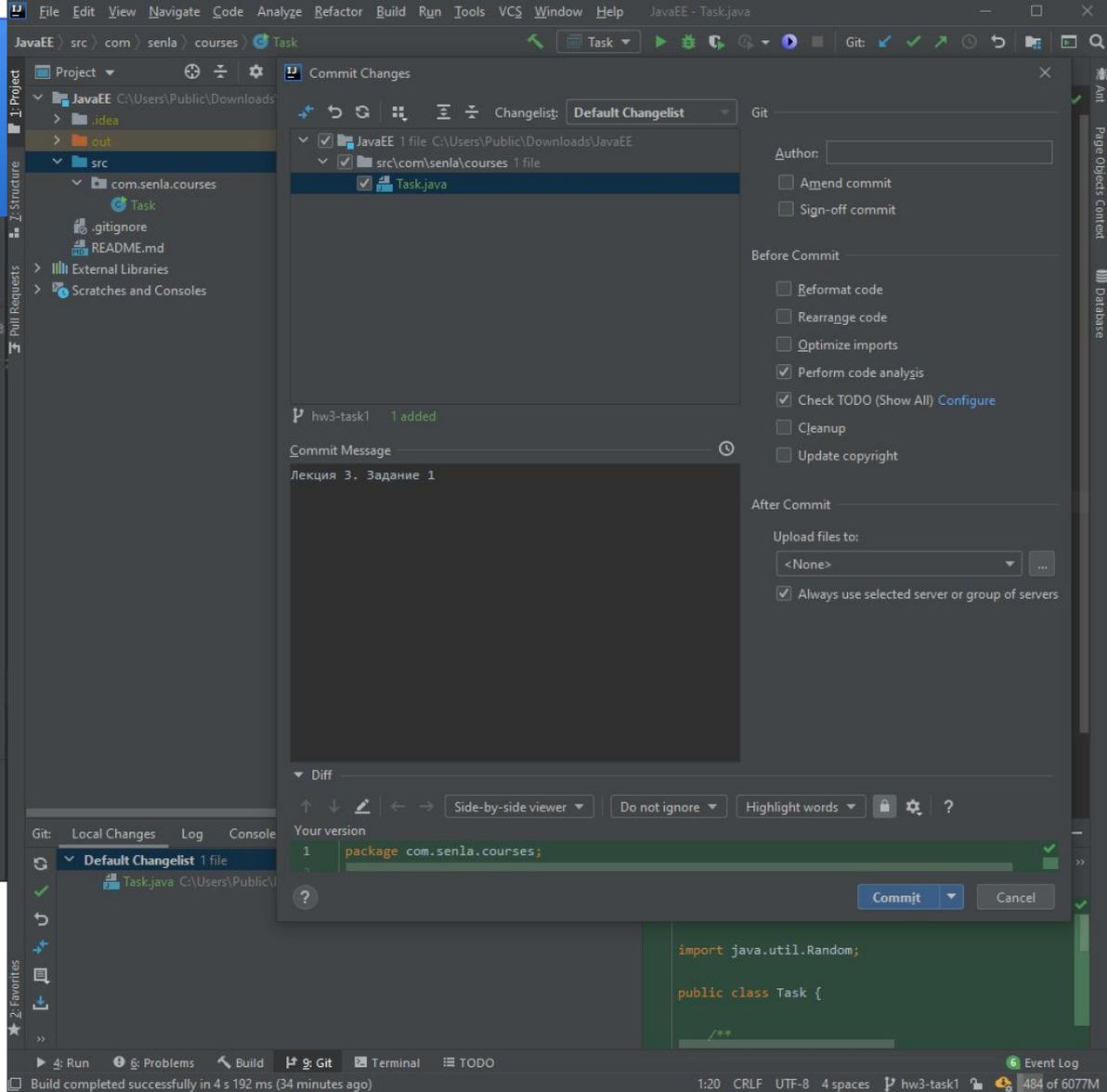
В итоге он попадет в список Default Changelist

Делаем коммит и пуш



При пуше видно, что отправляем изменения ветки hw3-task1 в origin : hw3-task1 (ветки в Удалённом репозитории ещё нет, и идея говорит что она будет создана New).

Из нового – только 1 коммит в списке ниже.



Проверяем на github

Search or jump to... | Pulls | Issues | Marketplace | Explore

JExample / JavaEE (Private) | Unwatch 1 | Star 0 | Fork 0

<> Code | Issues | Pull requests | Actions | Projects | Security | Insights | Settings

hw3-task1 had recent pushes 1 minute ago | [Compare & pull request](#)

hw3-task1 | Go to file | Add file | Code

This branch is 1 commit ahead of master. | Pull request | Compare

vvoronovich and vvoronovich Лекция 3. Задание 1 ... 4 minutes ago 3

File	Commit Message	Time
src/com/senla/courses	Лекция 3. Задание 1	4 minutes ago
.gitignore	Добавлены исключения в .gitignore .	2 hours ago
README.md	Initial commit	2 hours ago

README.md | JavaEE | Проект для курсов JavaEE

Search or jump to... | Pulls | Issues | Marketplace | Explore

JExample / JavaEE (Private) | Unwatch 1 | Star 0 | Fork 0

<> Code | Issues | Pull requests | Actions | Projects | Security | Insights | Settings

hw3-task1

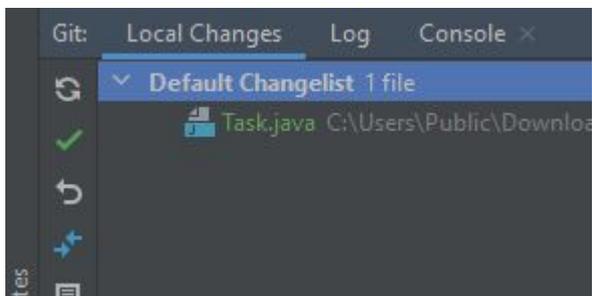
Commits on May 26, 2021

Лекция 3. Задание 1	vvoronovich authored and vvoronovich committed 3 minutes ago	3cbd517
Добавлены исключения в .gitignore .	vvoronovich authored and vvoronovich committed 2 hours ago	226ffc8
Initial commit	JExample committed 2 hours ago	2a75808 (Verified)

Newer Older

© 2021 GitHub, Inc. | Terms | Privacy | Security | Status | Docs | Contact GitHub | Pricing | API | Training | Blog | About

После создания класса с методом main и выполнения, создаётся исключенная ранее папка out.



The screenshot shows an IDE with the source code of a Java class named 'Task'. The code is as follows:

```
package com.senla.courses;

import java.util.Random;

public class Task {

    /**
     * Задание 1: Написать программу
     * <p>
     * p.s. больше никаких требований, поэтому можно просто внутри main написать весь код
     * <p>
     * выбрал пункт 5 (у вас его нет)
     * 5) выводящую на экран случайно сгенерированное целое число от -555 до 555 и сумму его
     * цифр
     */
    public static void main(String[] args) {
        //генерируем число от -555 до 555
        int randomNumber = new Random().nextInt(1111) - 555;
        System.out.println("random number = " + randomNumber);
        //конвертируем в положительное число
        if (randomNumber < 0) {
            randomNumber = -randomNumber;
        }
        //считаем сумму его цифр
        int sum = 0;
        do {
            sum += randomNumber % 10;
            randomNumber /= 10;
        }
        while (randomNumber != 0);
        System.out.println("sum = " + sum);
    }
}
```

The Run console shows the output of the program:

```
Run: Task
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
random number = -151
sum = 7
Process finished with exit code 0
```

Проверяем, что ветка master не изменила состояние

The screenshot shows the GitHub interface for a repository named 'JExample / JavaEE'. The repository is private and has 1 star and 0 forks. The 'Code' tab is selected, showing a list of files: '.gitignore' (added 2 hours ago) and 'README.md' (initial commit, 2 hours ago). The README content is visible, showing the title 'JavaEE' and the subtitle 'Проект для курсов JavaEE'. The right sidebar contains sections for 'About' (Project for JavaEE courses), 'Releases' (No releases published), and 'Packages' (No packages published). The footer includes copyright information for GitHub, Inc. and various links like Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

Search or jump to... / Pulls Issues Marketplace Explore

JExample / JavaEE Private Unwatch 1 Star 0 Fork 0

<> Code Issues Pull requests Actions Projects Security Insights Settings

hw3-task1 had recent pushes 3 minutes ago Compare & pull request

master Go to file Add file Code

vvoronovich and vvoronovich Добавлены исключения в .gitignore 2 hours ago 2

.gitignore Добавлены исключения в .gitignore . 2 hours ago

README.md Initial commit 2 hours ago

README.md

JavaEE

Проект для курсов JavaEE

About
Проект для курсов JavaEE

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

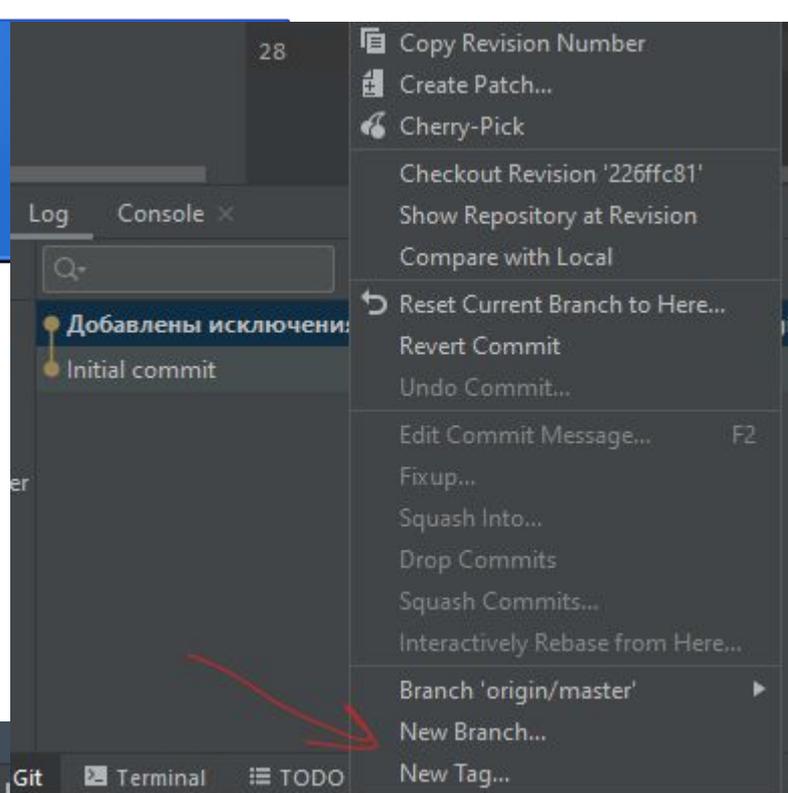
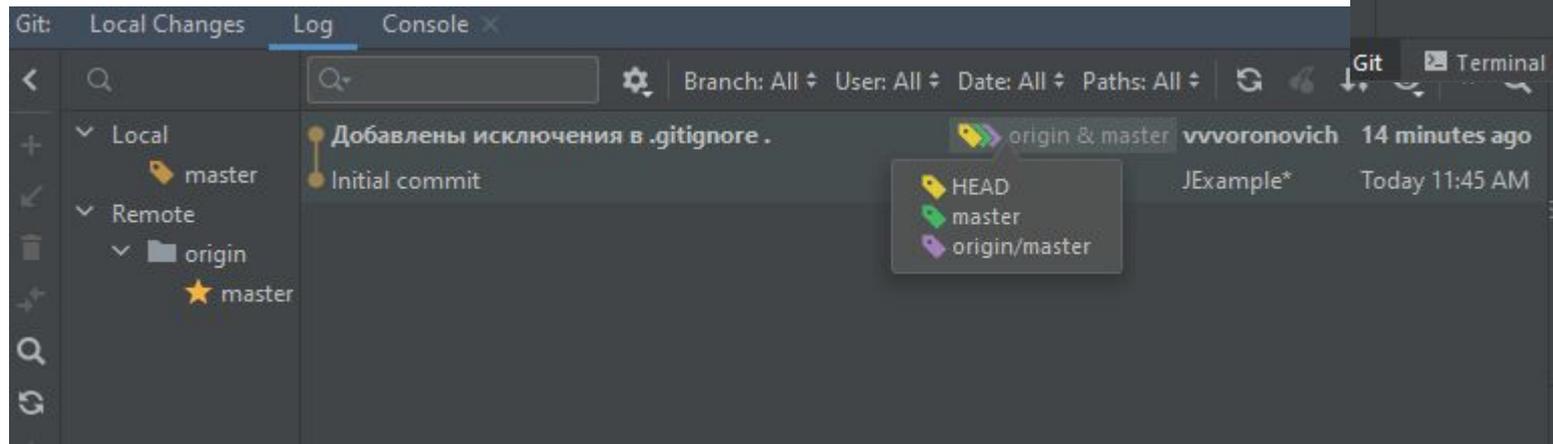
© 2021 GitHub, Inc. Terms Privacy Security Status Docs
Contact GitHub Pricing API Training Blog About

Ветвление в IDEA

Во вкладке Git есть раздел Log, в нем отображается дерево коммитов и маркируются ветки.

Создавать ветку в гите можно от любого коммита.

Поэтому при нажатии ПКМ на коммит из списка есть вариант New Branch...



Создаём ветку для следующего задания

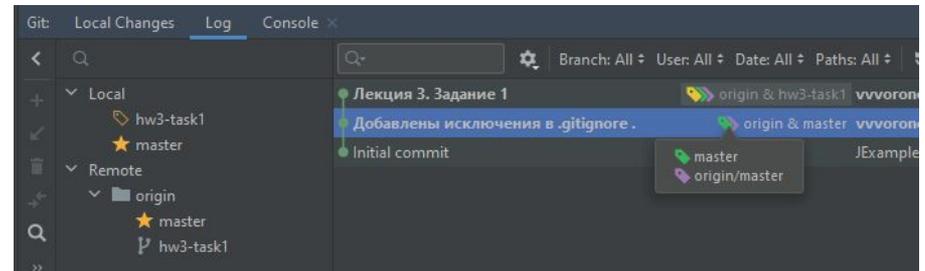
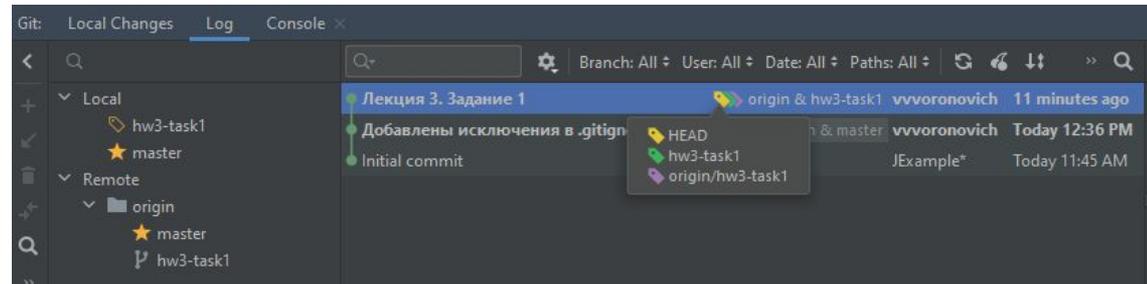
1) Можно сделать ветку от коммита (слайд 13)

2) Можно сделать ветку от ветки (слайд 16)

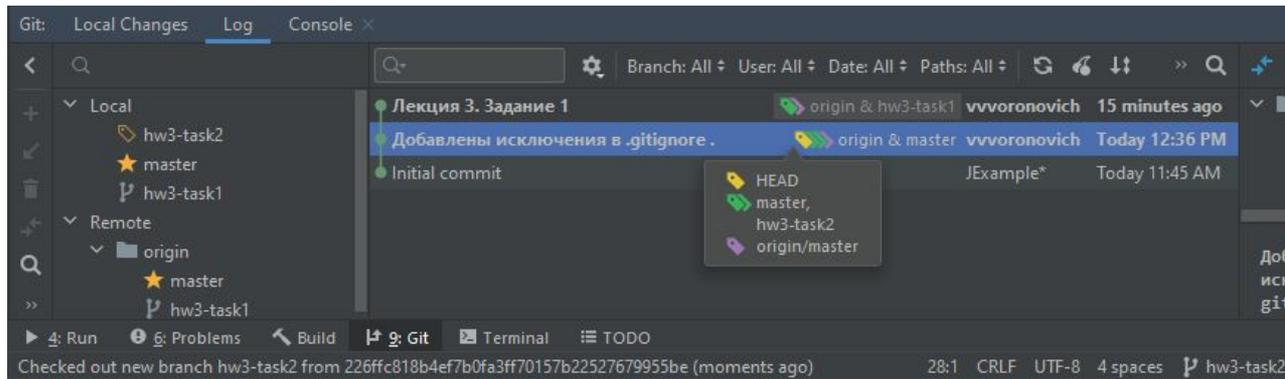
Ещё раз пройдём путём 1.

Если сравнить скриншоты, то видно, что первый сверху коммит это последний коммит в ветку hw3-task1.

Нас же интересует создание новой ветки от master, поэтому мы должны выбрать второй коммит, который является последним для master, и создать новую ветку hw3-task2

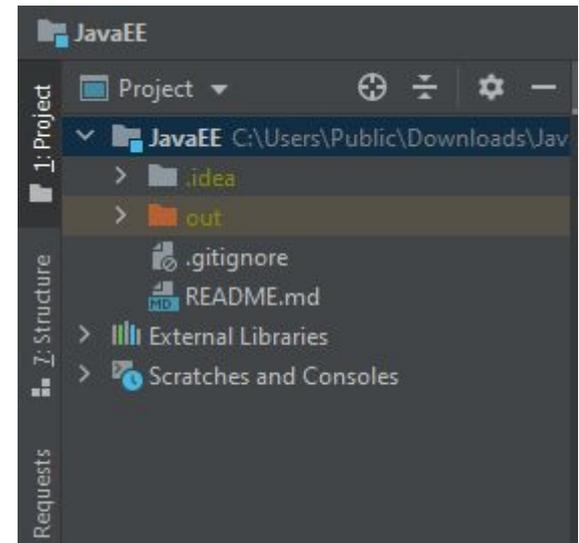


После создания ветки и присвоения ей названия



Видим что в списке появилась ветка hw3-task2, которую создавали на предыдущем слайде

И папка src и всё ее содержимое “пропало”



Выполнение задания 2

- 1) Повторяем создание папки src, как на слайде 17.
- 2) Создаём необходимые пакеты и решаем поставленную задачу.
- 3) Когда создаём новый файл помним про слайды 18-19.

Абстракция

```
.gitignore x AbstractFuel.java x
1 package com.senla.courses.fuel;
2
3 /**
4  * Класс, описывающий абстрактное топливо
5  */
6 public abstract class AbstractFuel {
7
8     /**
9      * Количество выделяемого CO2
10     */
11     private final int carbonEmission;
12
13     /**
14      * Обязательный конструктор абстрактного класса
15      *
16      * @param carbonEmission количество выделяемого CO2
17      */
18     protected AbstractFuel(int carbonEmission) {
19         this.carbonEmission = carbonEmission;
20     }
21
22     /**
23      * Получить количество выделяемого CO2
24      *
25      * @return количество выделяемого CO2
26      */
27     public int getCarbonEmission() {
28         return carbonEmission;
29     }
30
31     /**
32      * Метод для получения наименования топлива
33      *
34      * @return наименование топлива
35      */
36     public abstract String getFuelName();
37
38     /**
39      * Узнать, является ли топливо экологичным (не выделяет CO2)
40      */
41     public boolean isEcoFriendly() {
42         return carbonEmission == 0;
43     }
44 }
45
```

Первый наследник

```
.gitignore x AbstractFuel.java x Diesel.java x
1 package com.senla.courses.fuel;
2
3 /**
4  * Дизельное топливо
5  */
6 public class Diesel extends AbstractFuel {
7
8     /**
9      * Количество CO2 выделяемое при использовании дизельного топлива
10     */
11     private static final int DIESEL_FUEL_CARBON_EMISSION = 100;
12
13     /**
14      * Наименование дизельного топлива
15     */
16     private static final String DIESEL_FUEL_NAME = "Солярка";
17
18     /**
19      * Конструктор, который использует обязательный конструктор абстрактного класса
20     */
21     public Diesel() {
22         super(DIESEL_FUEL_CARBON_EMISSION);
23     }
24
25     @Override
26     public String getFuelName() {
27         return DIESEL_FUEL_NAME;
28     }
29 }
```

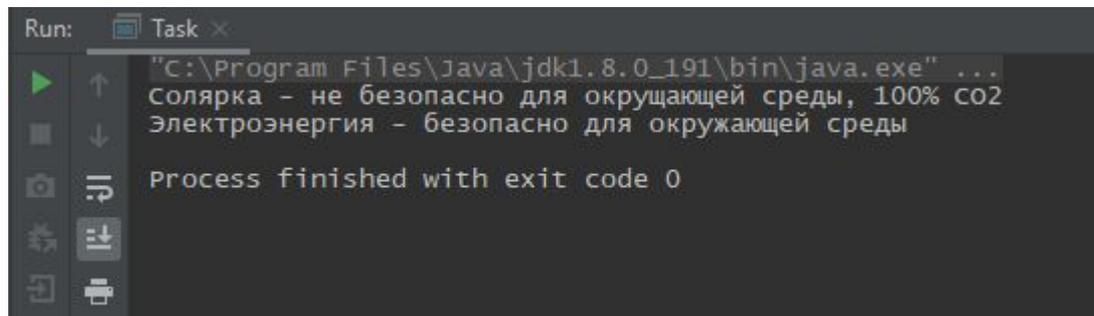
Второй наследник

```
.gitignore x AbstractFuel.java x Diesel.java x Electricity.java x
1 package com.senla.courses.fuel;
2
3 /**
4  * Электричество
5  */
6 public class Electricity extends AbstractFuel {
7
8     /**
9      * Количество CO2 выделяемое при использовании дизельного топлива
10     */
11     private static final int ELECTRICITY_CARBON_EMISSION = 0;
12
13     /**
14      * Названия электрического топлива
15     */
16     private static final String ELECTRICITY_FUEL_NAME = "Электроэнергия";
17
18     /**
19      * Конструктор, который использует обязательный конструктор абстрактного класса
20     */
21     public Electricity() {
22         super(ELECTRICITY_CARBON_EMISSION);
23     }
24
25     @Override
26     public String getFuelName() {
27         return ELECTRICITY_FUEL_NAME;
28     }
29 }
```

Проверка работы

```
.gitignore x AbstractFuel.java x Diesel.java x Electricity.java x Task.java x
1 package com.senla.courses;
2
3 import com.senla.courses.fuel.AbstractFuel;
4 import com.senla.courses.fuel.Diesel;
5 import com.senla.courses.fuel.Electricity;
6
7 public class Task {
8
9     /**
10      * Задание 2
11      * Написать программу (только одну из 3-х):
12      *
13      * (у вас такой нет)
14      * 4) Содержащую иерархию топлива. Создать все виды топлива и определить, какие из них являются
15      * безопасными для окружающей среды, а если они не безопасны, то показать количество выделяемого CO2.
16      */
17     public static void main(String[] args) {
18         AbstractFuel[] fuels = new AbstractFuel[]{new Diesel(), new Electricity()};
19
20         for (AbstractFuel fuel : fuels) {
21             System.out.print(fuel.getFuelName());
22             if (fuel.isEcoFriendly()) {
23                 System.out.println(" - безопасно для окружающей среды");
24             } else {
25                 System.out.println(" - не безопасно для окружающей среды, " + fuel.getCarbonEmission() + "% CO2");
26             }
27         }
28     }
29 }
30
```

Вывод в консоль



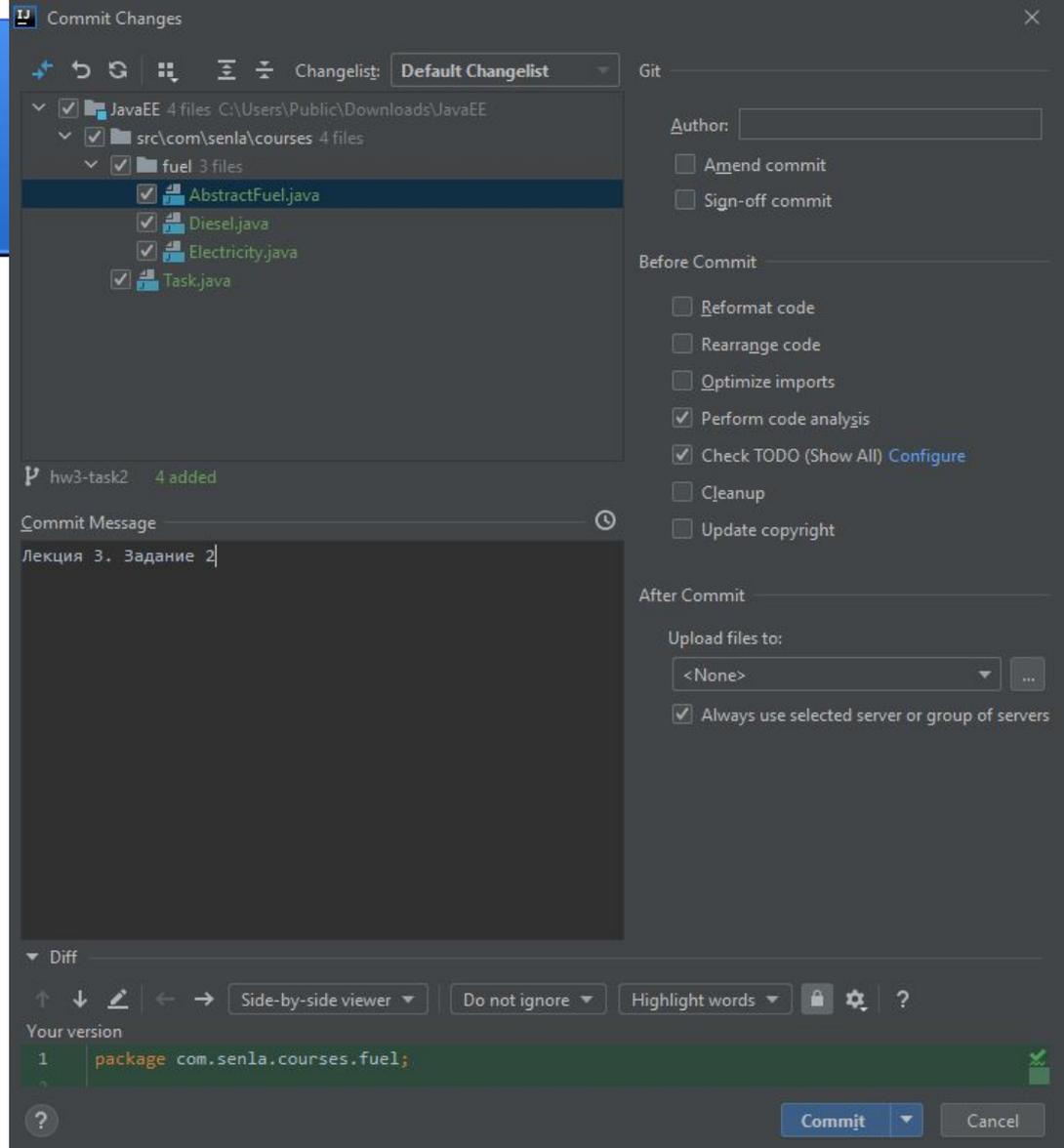
```
Run: Task x
"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
Солярка - не безопасно для окружающей среды, 100% CO2
Электроэнергия - безопасно для окружающей среды

Process finished with exit code 0
```

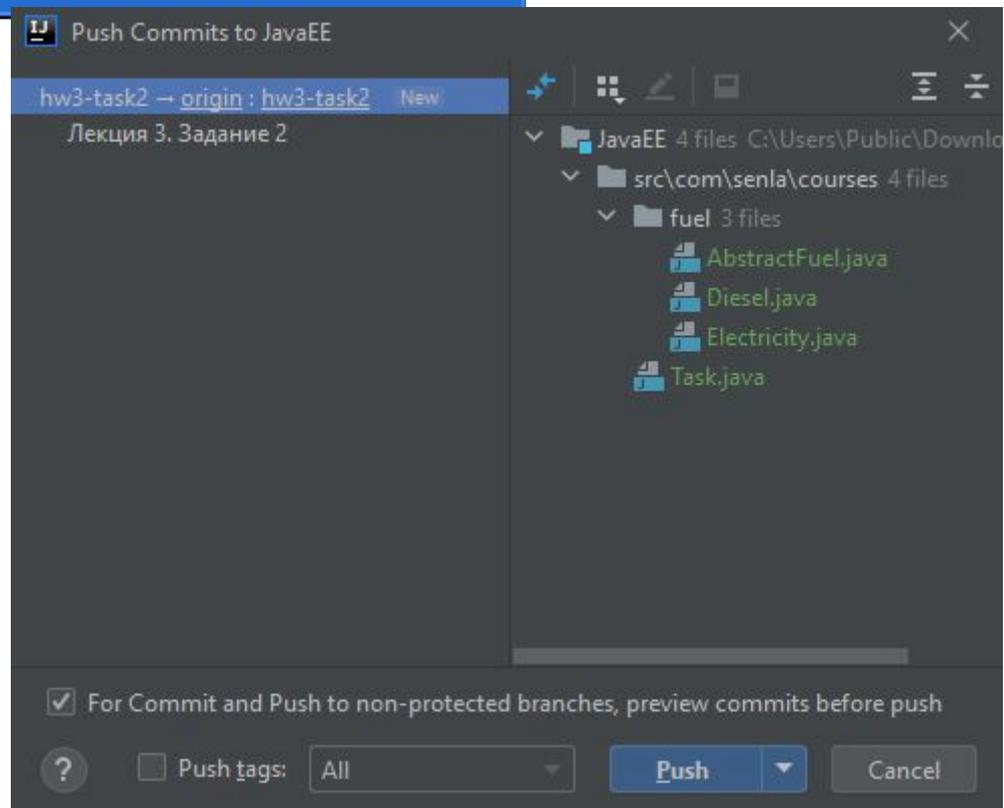
Делаем новый КОММИТ

Меняем текст сообщения, убеждаемся что в списке сверху все необходимые файлы.

Делаем Commit и Push



- Пуш из ветки hw3-task2 в новую ветку origin : hw3-task2
- Из изменений только 1 наш коммит.



Проверяем на github

The screenshot displays the GitHub interface for the repository `JExample / JavaEE`. The top navigation bar includes links for Pulls, Issues, Marketplace, and Explore. The repository name is shown as `JExample / JavaEE` with a 'Private' label. The main content area is divided into several sections:

- Recent Commits:** Two commits are listed: `hw3-task1` (pushed about 1 hour ago) and `hw3-task2` (pushed 2 minutes ago). Each has a 'Compare & pull request' button.
- File Browser:** A dropdown menu shows `hw3-task2`. Below it are buttons for 'Go to file', 'Add file', and 'Code'.
- Branch Status:** A message states 'This branch is 1 commit ahead of master.' with links for 'Pull request' and 'Compare'.
- Commit History:** A list of commits by `vvoronovich`:
 - `src/com/senla/courses`: Лекция 3. Задание 2 (3 minutes ago)
 - `.gitignore`: Добавлены исключения в .gitignore (2 hours ago)
 - `README.md`: Initial commit (3 hours ago)
- README.md:** A preview of the `README.md` file showing the title `JavaEE` and the subtitle 'Проект для курсов JavaEE'.

The right sidebar contains sections for 'About' (Project for JavaEE courses), 'Releases' (No releases published), and 'Packages' (No packages published).

ИТОГ

Мы имеем 3 ветки

1) master – которую мы пока не меняем, но в ней есть полезный gitignore, поэтому для создания новых веток мы используем ее в качестве базы.

2) hw3-task1 созданная на основе master, и содержащая свой коммит с решением задачи 1

3) hw3-task2 созданная на основе master, и содержащая свой коммит с решением задачи 2

Для всех задач 3 лекции используем отдельные ветки. В идее можно переключаться между ними с помощью checkout, и гит будет подменять содержимое папки

The screenshot shows the GitHub interface for a repository named 'JExample / JavaEE'. The repository is private and has 1 pull request, 0 stars, and 0 forks. The 'All branches' tab is selected, showing a list of branches: 'master' (Default), 'hw3-task2', and 'hw3-task1'. A context menu is open over the 'hw3-task1' branch, showing options like 'Checkout', 'New Branch from Selected...', 'Checkout and Rebse onto Current', 'Compare with Current', 'Show Diff with Working Tree', 'Rebase Current onto Selected', 'Merge into Current', 'Update', 'Push...', 'Rename...', and 'Delete'. The 'Checkout' option is highlighted.