



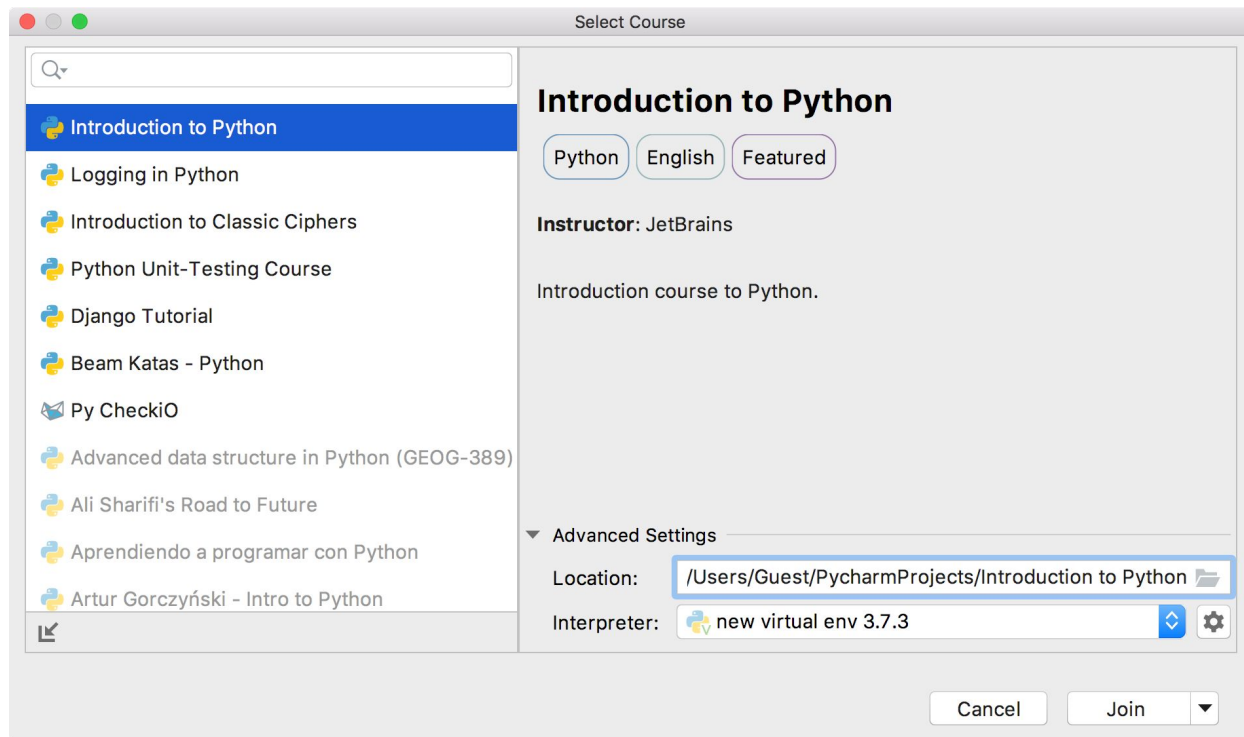
JetBrains Educational Products

Санкт-Петербург, TechTrain, 2019

Educational Products - ЧТО ЭТО?

О наших инструментах

- Мы - платформа
- В основе наших решений - зарекомендовавшие себя IDE
- Мы стремимся к автоматизации



Educational Products для учащихся

jb.gg/learn



Обучение в профессиональной среде

The screenshot displays an IDE window for a Python project titled "Introduction to Python28". The main editor shows the following code in `return_keyword.py`:

```
8 def fib(n): n: 10
9     """This is documentation string
10    Return a list containing the Fibonacci sequence up to n"""
11    result = []
12    a = 1
13    b = 1
14    while b < 1000000:
15        a, b = b, a + b
16    return result
```

The function `fib()` is currently being debugged. The debugger interface at the bottom shows the following state:

- Debugger:** `return_keyword`
- Frames:** `Mai.....`, `fib, return_keyword.`, `<module>, return_k`
- Variables:**
 - `a = {int} 1`
 - `b = {int} 1000000`
 - `n = {int} 10`
 - `result = {list} <class 'list': []`

On the right side, the **Task Description** panel contains the following text:

assign it to a variable or just print it out.

In the Fibonacci sequence, the first two numbers are 1 and 1, and each subsequent number is the sum of the previous two. Write a function that returns a list of the numbers of

A **Check** button is visible below the task description.

At the bottom of the IDE, the status bar shows: Frameworks Detected: Python framework is detected. // Configure (a minute ago) 17:1 LF UTF-8 4 spaces

Простор для экспериментов

The image shows a screenshot of an IDE (likely VS Code) with a project named "JavaScript for Beginners". The main editor window displays the file "example.html" with the following code:

```
1 <html>
2   <head>
3     <title>Code sample page with JavaScript</title>
4     <script>
5       alert("Hello JavaScript!");
6     </script>
7   </head>
8   <body>
9     <p>This is the main page text</p>
10  </body>
```

On the right side, a "Task Description" panel contains the following text:

You can **implement JavaScript in an HTML document** in two ways:

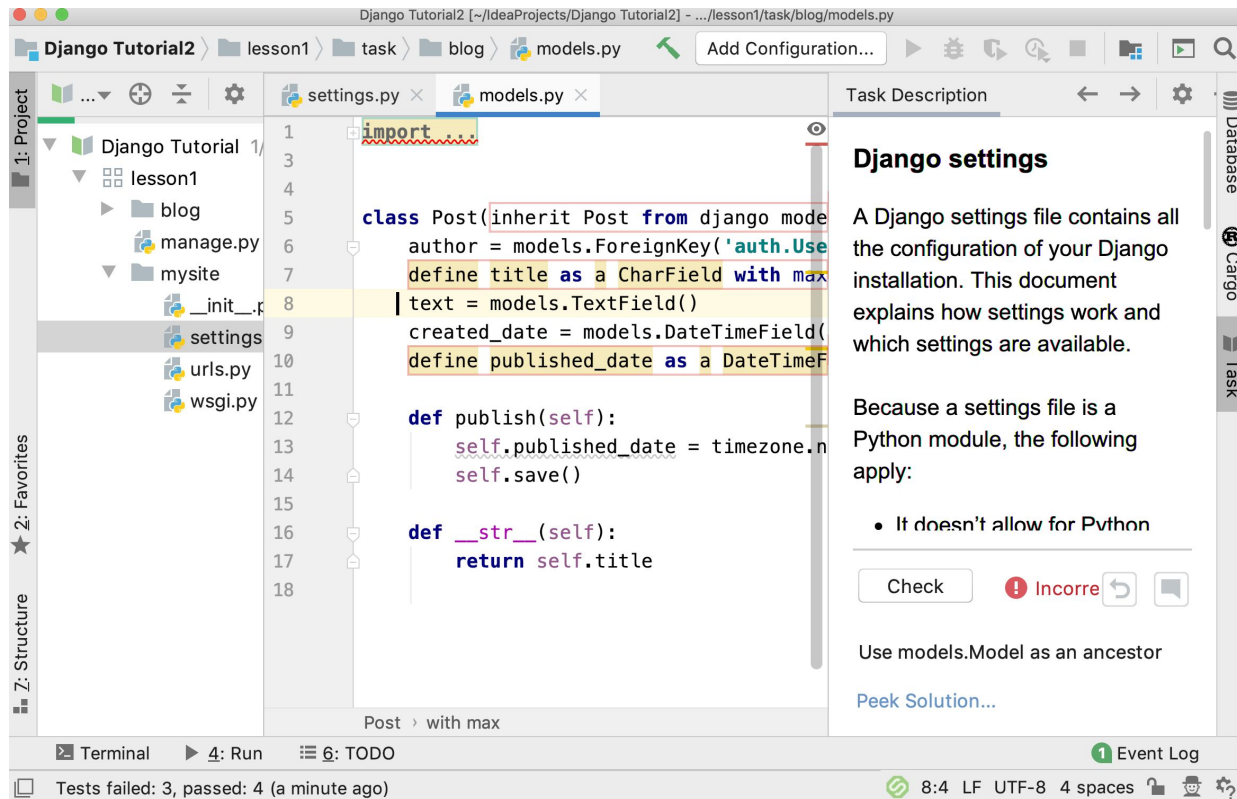
The first is to place the code directly **in HTML file**

The second is to place the code **in a separate file**.

Below the IDE, a browser window (Incognito) is open at "localhost:63343/JavaScript_for_Beginners2...". It displays an alert dialog box with the text "Hello JavaScript!" and an "OK" button.

Проектно-ориентированное обучение

- Умение декомпозировать и решать задачу.
- Последовательная доработка первого решения



The screenshot shows an IDE window titled "Django Tutorial2" with the following structure:

- Project: Django Tutorial 1
 - lesson1
 - blog
 - manage.py
 - mysite
 - __init__.py
 - settings.py
 - urls.py
 - wsgi.py

The code editor displays the content of `models.py`:

```
1 import ...
2
3
4
5 class Post(inherit Post from django mode
6     author = models.ForeignKey('auth.Use
7     define title as a CharField with max
8     text = models.TextField()
9     created_date = models.DateTimeField(
10    define published_date as a DateTimeF
11
12 def publish(self):
13     self.published_date = timezone.n
14     self.save()
15
16 def __str__(self):
17     return self.title
18
```

The right sidebar shows the "Task Description" for "Django settings":

Django settings

A Django settings file contains all the configuration of your Django installation. This document explains how settings work and which settings are available.

Because a settings file is a Python module, the following apply:

- It doesn't allow for Python

Buttons: Check, ! Incorre, ↶, 💬

Use models.Model as an ancestor

[Peek Solution...](#)

Terminal: 4: Run, 6: TODO

Event Log: 1

Tests failed: 3, passed: 4 (a minute ago)

8:4 LF UTF-8 4 spaces

JetBrains Academy

jetbrains.com/academy

JetBrains Academy: проектное обучение

Project: Simple Chatty Bot

Easy ⌚ 9 hours

Learners who successfully completed this project rated it as follows (based on 1122 reviews):

Usefulness: 4.2 / 5

Clarity: 4.2 / 5

Fun: 4.2 / 5

About:

Here, at the beginning of your programmer's path, you will need a console bot that will guide you through the basics of software development. During this journey you will also play some word and number games that you are going to implement by yourself. Pack up and hit the road, my friend!

Result:

As you move through the project, you'll install Java on your machine and learn to work in a professional development environment. You'll get to know the basic syntax of Java and write a simple program using variables, conditions, loops, and methods.

Select this project

What you'll do and what you'll learn:

Stage 1/4: Hello! What's your name?

Teach your assistant to introduce itself in the console

Introduction to Java ...

Basic literals ...

The first program ...

Printing data ...

<https://hi.hyperskill.org/> - платформа по проектно-ориентированному обучению Java.

JetBrains Academy: теоретическая база

Java

Basic syntax and simple programs

Introduction to Java ...

Basic literals ...

The first program ...

Printing data ...

Types and variables ...

Primitive and reference types ...

Array ...

Comments ...

Naming variables ...

Learn next

Environment and tools

Write, compile and run ...

JVM, JRE and JDK ...

Build tools ...

Running programs on your computer ...

Maven basics ...

Gradle basics ...

Java Archive ...

The simplest project with Gradle ...

Dependency ...

Collections

What are collections ...

Basics of collections ...

Queue and Stack ...

ArrayList ...

List ...

Set ...

Map ...

The Collections Framework overview ...

The utility class Collections ...

Implementation of basic algorithms

Linear search in Java ...

Binary search in Java ...

Jump search in Java ...

Selection sort in Java ...

Bubble sort in Java ...

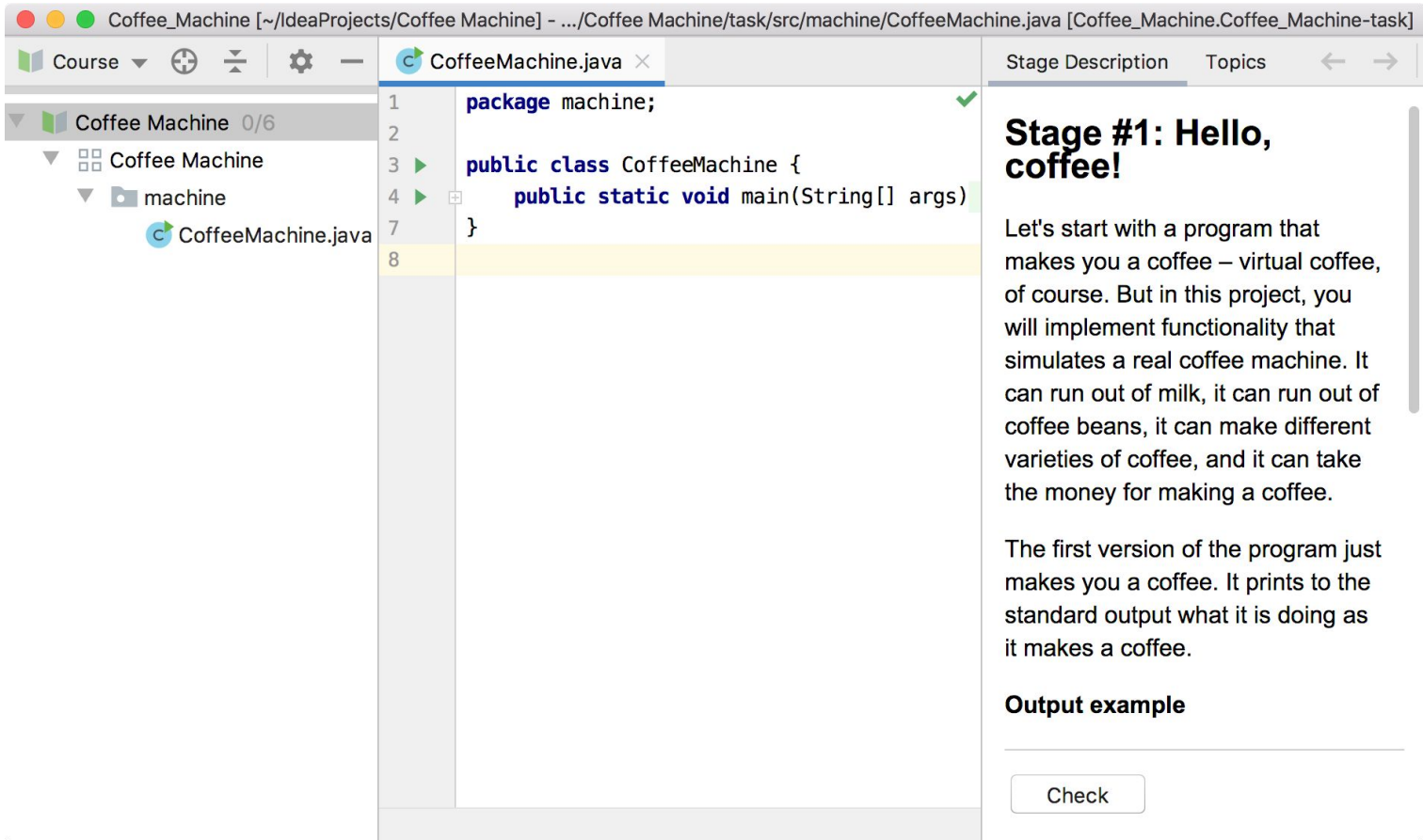
Insertion sort in Java ...

Counting sort in Java ...

Merge sort in Java ...

Quicksort in Java ...

JetBrains Academy: Интеграция с IDE



The screenshot shows the JetBrains Academy IDE interface. The top bar displays the project name "Coffee_Machine" and the file path. The left sidebar shows a project tree with "Coffee Machine" and "machine" folders, and the "CoffeeMachine.java" file selected. The main editor window shows the following Java code:

```
1 package machine;
2
3 public class CoffeeMachine {
4     public static void main(String[] args)
5     {
6
7     }
8
```

The right sidebar contains a task description for "Stage #1: Hello, coffee!". The description text is:

Stage #1: Hello, coffee!

Let's start with a program that makes you a coffee – virtual coffee, of course. But in this project, you will implement functionality that simulates a real coffee machine. It can run out of milk, it can run out of coffee beans, it can make different varieties of coffee, and it can take the money for making a coffee.

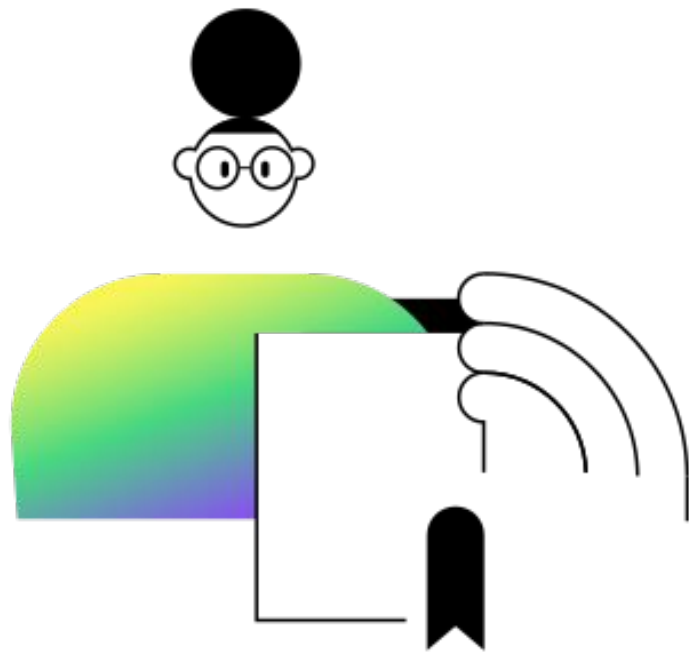
The first version of the program just makes you a coffee. It prints to the standard output what it is doing as it makes a coffee.

Output example

At the bottom of the right sidebar, there is a "Check" button.

Educational Products для преподавателей

jb.gg/teach



Для преподавателей

The screenshot shows an IDE window for a Kotlin project. The breadcrumb path is "Programming Basics > Functions > Exercise 1 > src > Task.kt". The editor displays the following code:

```
1 package functions1
2
3 fun getSquare(i: Int): Int = TODO()
4
5 fun main(args: Array<String>) {
6     println(getSquare(2)) // 4
7 }
```

The task description on the right is titled "Functions (#1)" and reads: "Create a function getSquare() that takes an Int argument and returns its square." Below the description is a "Check" button and a refresh icon.

The left sidebar shows the project structure:

- 1: Project
 - AtomicKotlin 3/224
 - Programming Basics
 - Hello, World!
 - var & val
 - Data Types
 - Functions
 - Examples
 - Exercise 1
 - Task.kt
- 2: Favorites
- Z: Structure
 - Exercise 2
 - Exercise 3
 - Exercise 4
 - if Expressions
 - String Templates
 - Number Types
 - Booleans
 - Repetition with whi
 - Looping & Ranges
 - The in Keyword

The bottom status bar shows "3 processes running...", "5:15 LF UTF-8 4 spaces", and a red error icon.

Для тех, кто делится опытом

The image shows a screenshot of an IDE window titled "Beam Katas - Python". The main editor displays the file "task.py" with the following code:

```
10 #
11 # Unless required by applicable
12 # distributed under the License
13 # WITHOUT WARRANTIES OR CONDITIO
14 # See the License for the specif
15 # limitations under the License.
16
17 import apache_beam as beam
18
19 from log_elements import LogElemen
20
21 p = beam.Pipeline()
22
23 (p | TODO())
24 | LogElements()
25
26 p.run()
27
```

The code is partially obscured by a "Task Description" panel on the right. The panel title is "Task Description" and the content is:

Hello Beam Pipeline

Apache Beam is an open source, unified model for defining both batch and streaming data-parallel processing pipelines. Using one of the open source Beam SDKs, you build a program that defines the pipeline. The pipeline is then executed by one of Beam's supported distributed processing back-ends, which include Apache Apex, Apache Flink, Apache Spark, and Google Cloud Dataflow.

At the bottom of the task description panel, there is a "Check" button and a refresh icon.

The IDE interface includes a project structure view on the left showing "Beam Katas - Python" with subfolders "Introduction", "Hello Beam", and "task.py". The status bar at the bottom indicates "Indexing..." and "23:12 LF UTF-8 4 spaces".

Demo JBQuizBot

JBQuizBot: Origins

@JBQuizBot предлагает вам
поучаствовать в викторине на
знание языков
программирования



JBQuizBot

Привет! Это викторина на знание языков программирования. Вам будут показаны 12 различных изображений и предложены варианты ответов. Для прохождения задания необходимо набрать 10 баллов, балл дается за каждый правильный ответ. Количество попыток неограничено. Для перезапуска нажмите на кнопку "Начать заново" или введите команду `/restart`.

Начать заново

```
function testRegex(s, sub_s) {
  let result = [];
  let pattern = new RegExp(sub_s, 'g');
  while (true) {
    let currentMatch = pattern.exec(s);
    if (currentMatch != null) result.push(currentMatch.toString());
    else break;
  }
  return result.join(' ');
}
```

1. Что бы это могло быть?

Судя по точкам с запятыми, Java

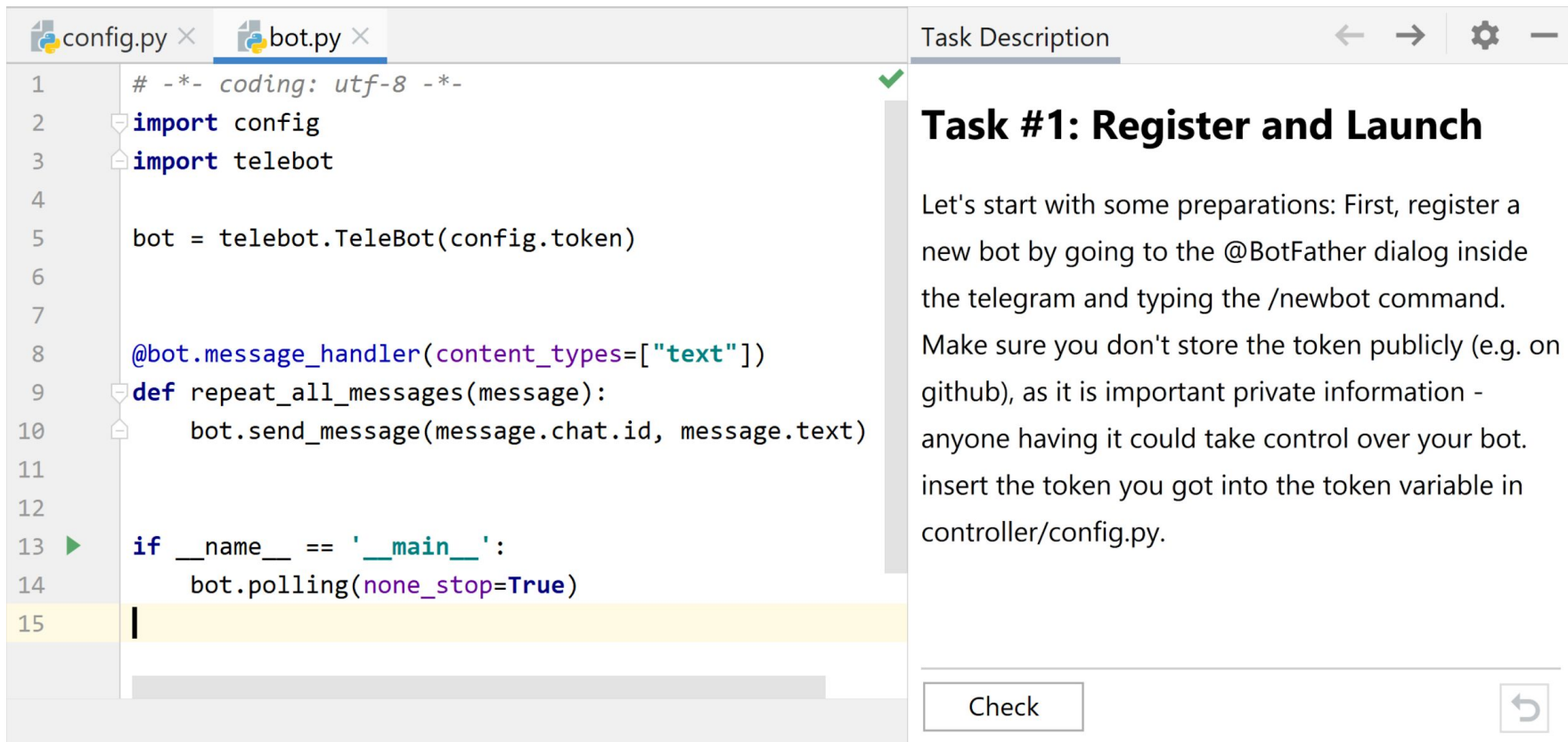
Да это же JS

Выглядит просто, это Python

Кратко, наверняка, Kotlin

Выглядит сложно, наверное, Scala

JVQuizBot: Первое сообщение и конфигурация



The image shows a code editor with two tabs: 'config.py' and 'bot.py'. The 'bot.py' tab is active and contains the following Python code:

```
1  # -*- coding: utf-8 -*-
2  import config
3  import telebot
4
5  bot = telebot.TeleBot(config.token)
6
7
8  @bot.message_handler(content_types=["text"])
9  def repeat_all_messages(message):
10     bot.send_message(message.chat.id, message.text)
11
12
13  if __name__ == '__main__':
14     bot.polling(none_stop=True)
15  |
```

A green checkmark is visible in the right margin of the code editor.

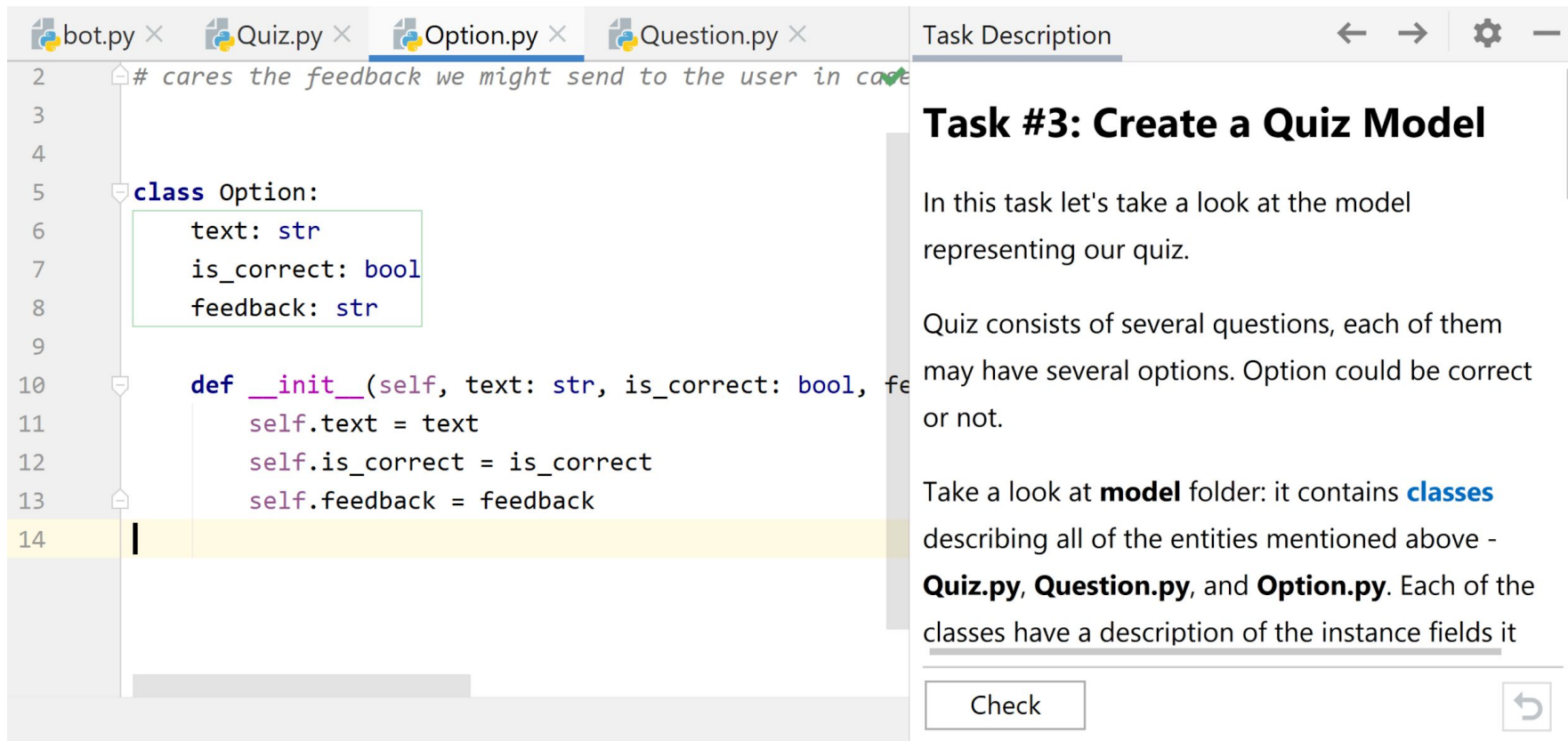
On the right side, there is a 'Task Description' panel with the following content:

Task #1: Register and Launch

Let's start with some preparations: First, register a new bot by going to the @BotFather dialog inside the telegram and typing the /newbot command. Make sure you don't store the token publicly (e.g. on github), as it is important private information - anyone having it could take control over your bot. insert the token you got into the token variable in controller/config.py.

At the bottom of the task description panel, there is a 'Check' button and a refresh icon.

JVQuizBot: Строим модель и контроллер



The image shows a code editor with four tabs: bot.py, Quiz.py, Option.py, and Question.py. The Option.py tab is active, displaying the following Python code:

```
2 # cares the feedback we might send to the user in case
3
4
5 class Option:
6     text: str
7     is_correct: bool
8     feedback: str
9
10 def __init__(self, text: str, is_correct: bool, fe
11     self.text = text
12     self.is_correct = is_correct
13     self.feedback = feedback
14
```

The code is partially highlighted with a yellow background. A green box highlights the class attributes, and a white box highlights the class definition. A vertical scrollbar is visible on the right side of the code editor.

On the right side, there is a panel titled "Task Description" with navigation arrows and a settings icon. The panel contains the following text:

Task #3: Create a Quiz Model

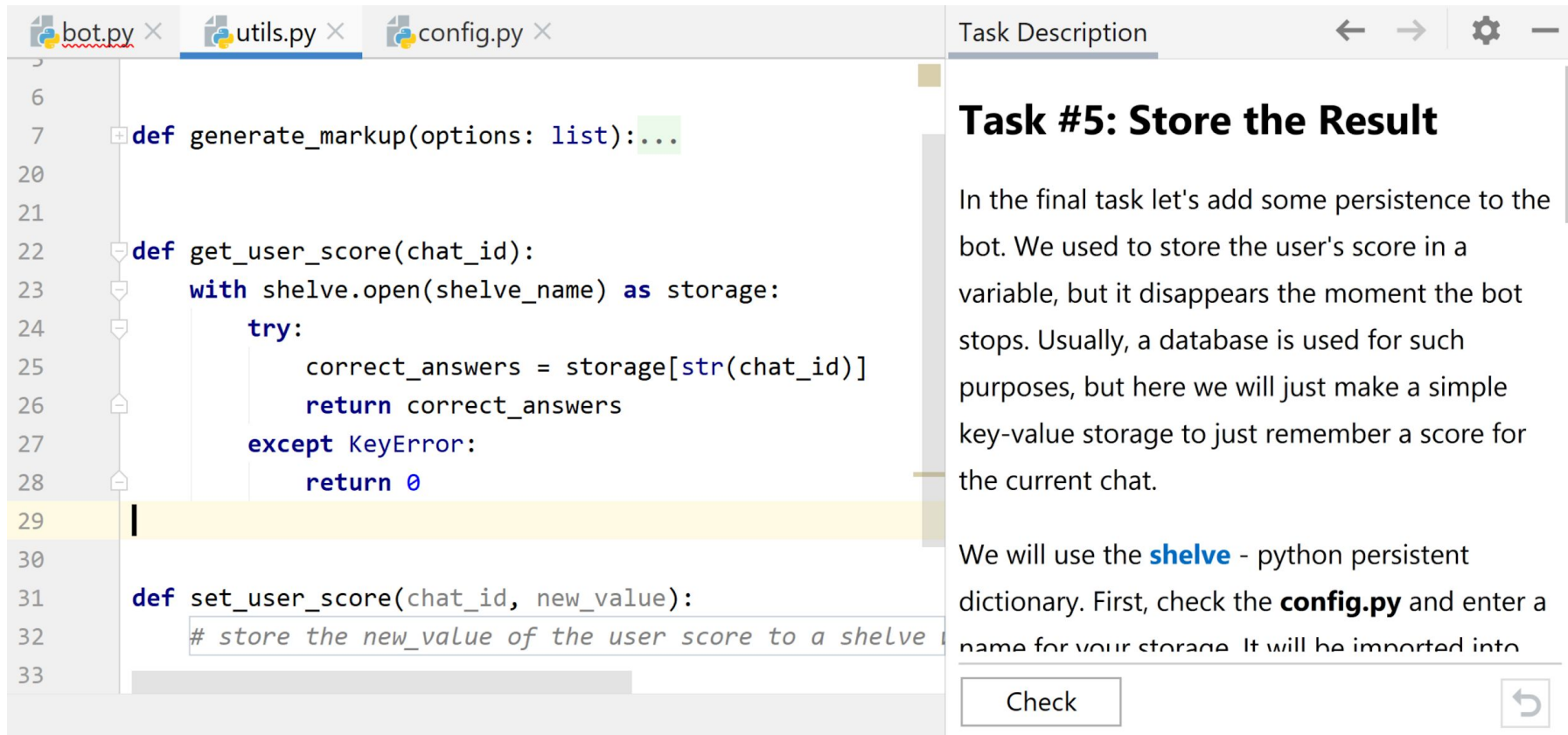
In this task let's take a look at the model representing our quiz.

Quiz consists of several questions, each of them may have several options. Option could be correct or not.

Take a look at **model** folder: it contains **classes** describing all of the entities mentioned above - **Quiz.py**, **Question.py**, and **Option.py**. Each of the classes have a description of the instance fields it

At the bottom of the panel, there is a "Check" button and a refresh icon.

JBQuizBot: Получаем интерактивное how-to



The image shows a code editor with three tabs: `bot.py`, `utils.py`, and `config.py`. The `utils.py` tab is active, displaying the following Python code:


```
5
6
7 def generate_markup(options: list):...
20
21
22 def get_user_score(chat_id):
23     with shelve.open(shelve_name) as storage:
24         try:
25             correct_answers = storage[str(chat_id)]
26             return correct_answers
27         except KeyError:
28             return 0
29
30
31 def set_user_score(chat_id, new_value):
32     # store the new_value of the user score to a shelve
33
```

On the right side, there is a "Task Description" panel with the following content:

Task #5: Store the Result

In the final task let's add some persistence to the bot. We used to store the user's score in a variable, but it disappears the moment the bot stops. Usually, a database is used for such purposes, but here we will just make a simple key-value storage to just remember a score for the current chat.

We will use the **shelve** - python persistent dictionary. First, check the **config.py** and enter a name for your storage. It will be imported into

Check 



Спасибо!

igor.gerasimov@jetbrains.com

[@JetBrains_Edu](https://www.jetbrains.com/idea/education/)

jetbrains.com/academy

jb.gg/learn

jb.gg/teach