

Справочник по работе с JSP-страницами (IntelliJ IDEA)

Существует множество интегрированных сред разработки (ИСР) подходящих для наших задач, но наиболее популярные – это *Eclipse* и *IntelliJ IDEA* от *JetBrains*. Мы рассмотрим вторую.

Для скачивания ИСР *IntelliJ IDEA* нужно перейти по данной ссылке: <https://www.jetbrains.com/idea/download/>

Для работы с JSP-страницами необходимо скачать *Ultimate* версию. Если вы студент или преподаватель, то вы можете получить ее бесплатно, подробности можно найти здесь:

<http://confluence.newit.gsu.by/display/dldoc/Student+Developer+Pack>

После скачивания запускаем скачанный исполняемый файл. Открывается окно, которое показано на рисунке 1.

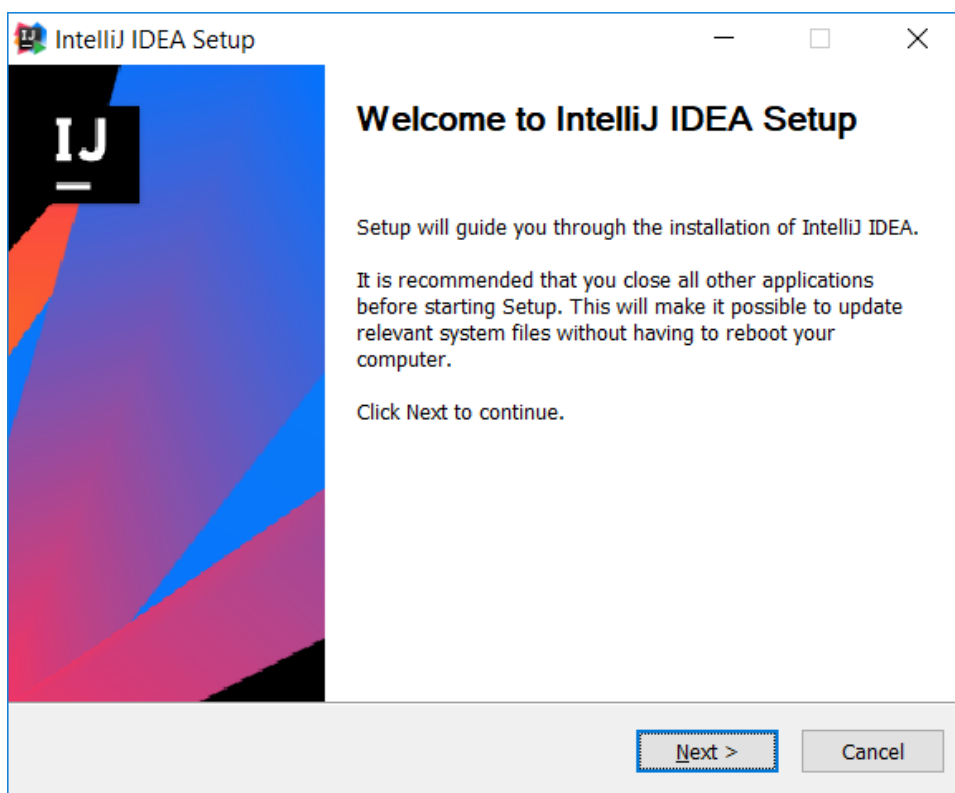


Рисунок 1 – Начало установки

Выбираем путь установки (рисунок 2).

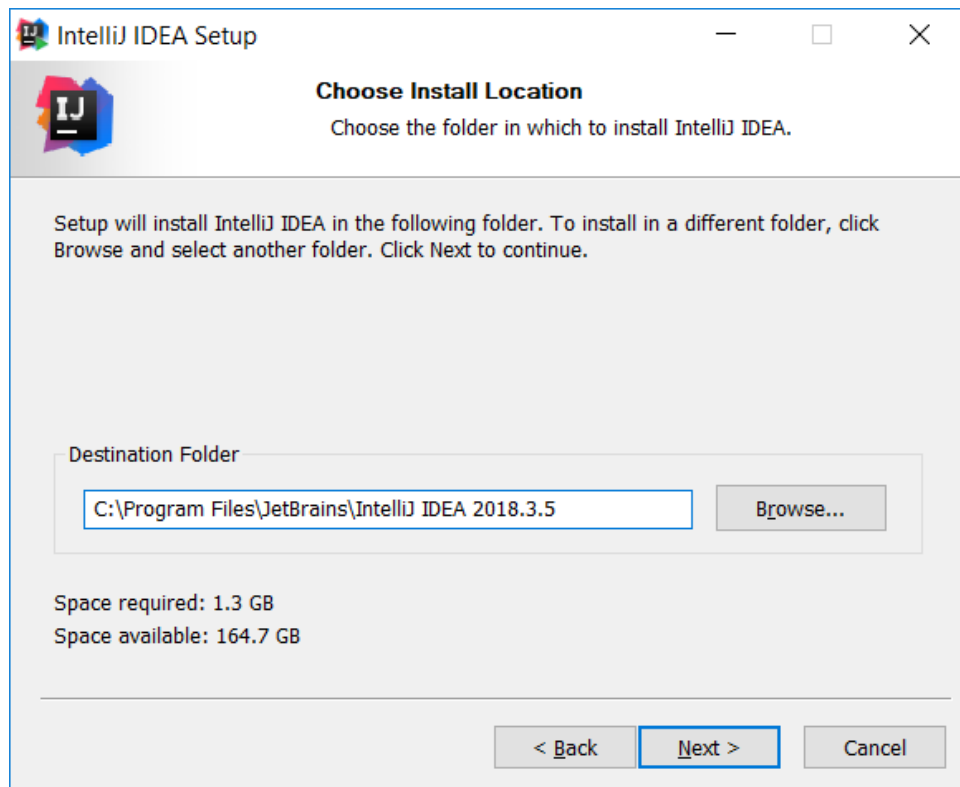


Рисунок 2 – Выбор установочного пути

Выбираем дополнительные параметры, как показано на рисунке 3.

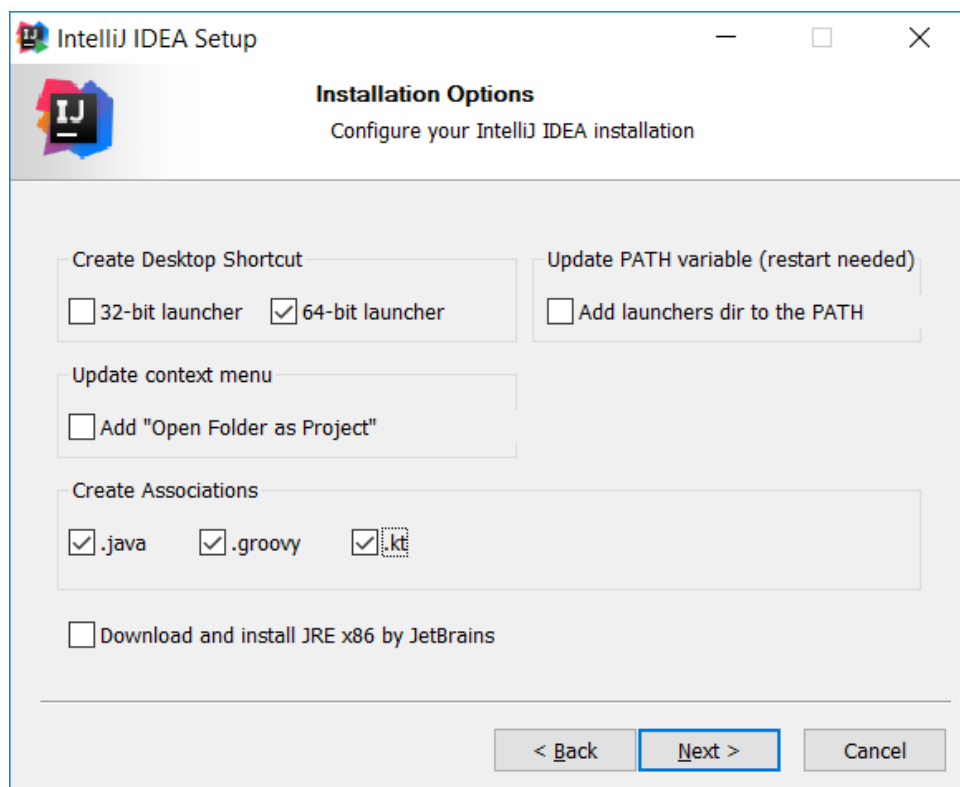


Рисунок 3 – Выбор дополнительных параметров

Устанавливаем ИСР (рисунок 4).

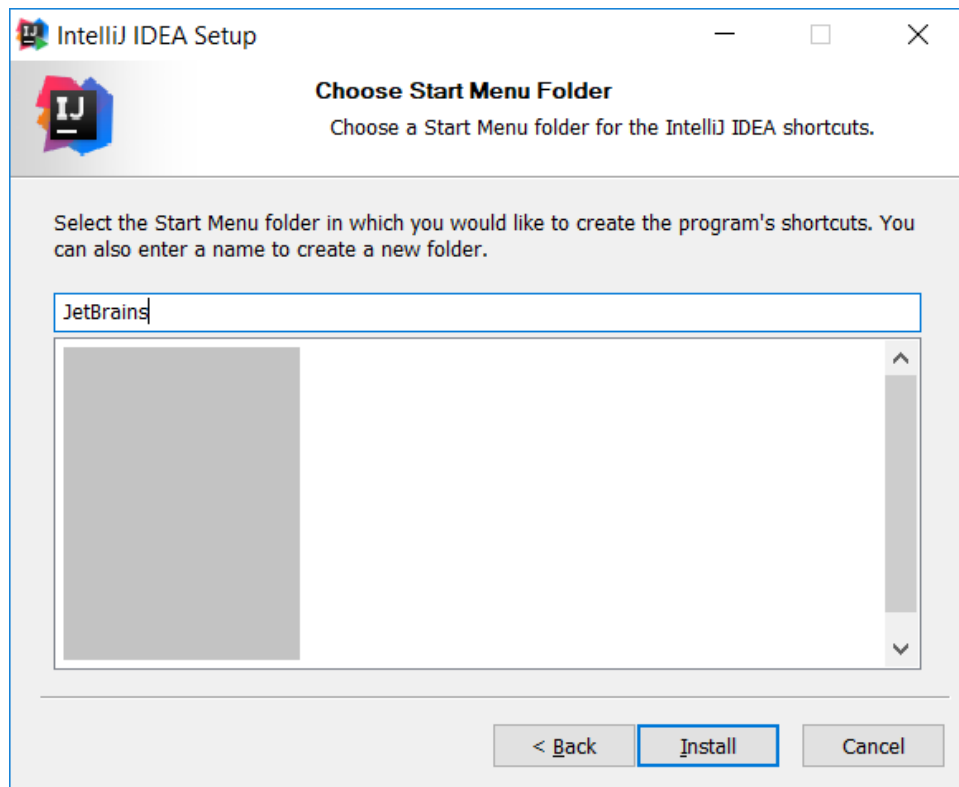


Рисунок 4 – Установка

После установки появится окно, как на рисунке 5. Устанавливаем флажок для запуска *IntelliJ IDEA* и нажимаем *Finish*.

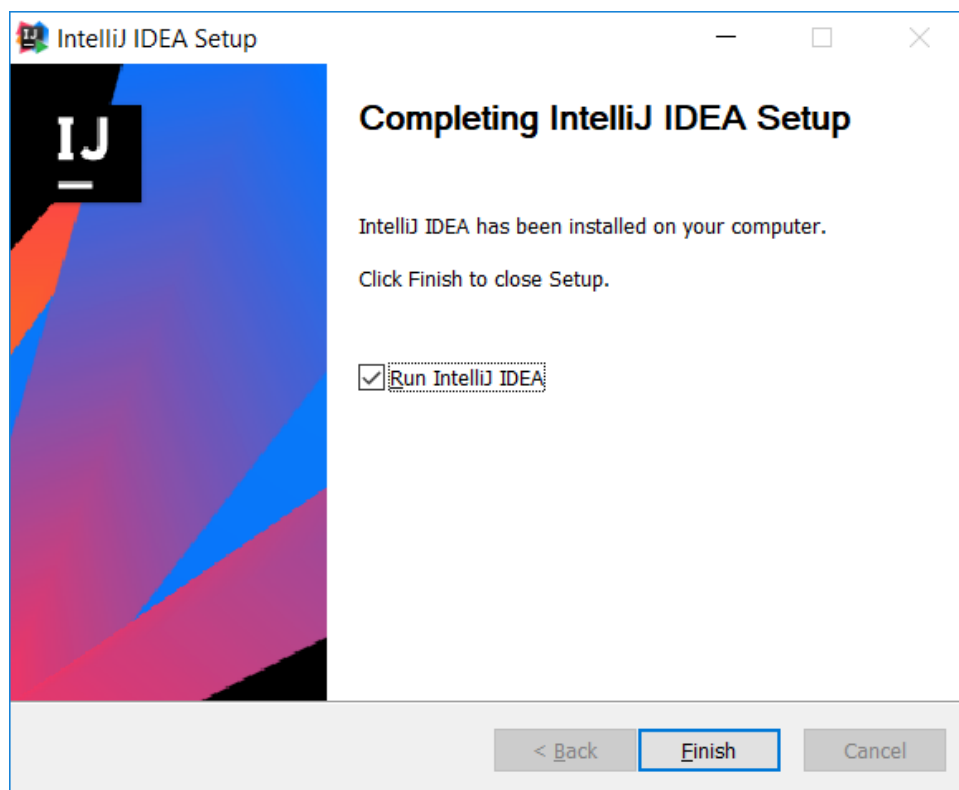


Рисунок 5 – Запуск

Предлагается импортировать настройки (рисунок 6). Мы устанавливаем впервые, следовательно, отказываемся.

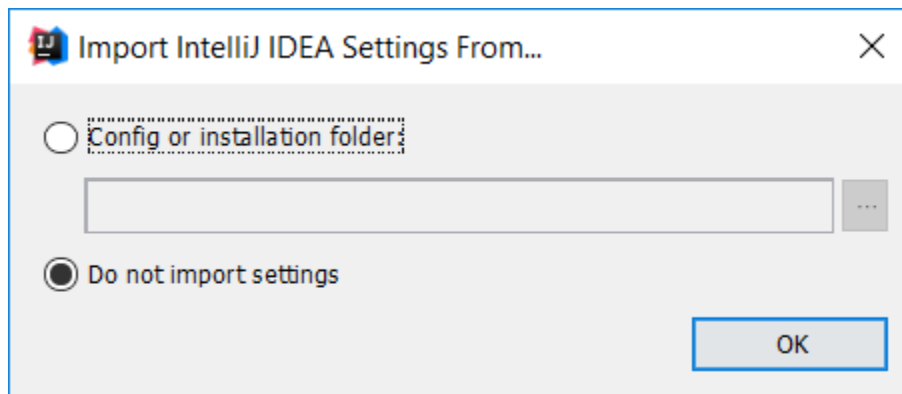


Рисунок 6 – Импорт настроек

Подтверждаем лицензионное соглашение (рисунок 7).

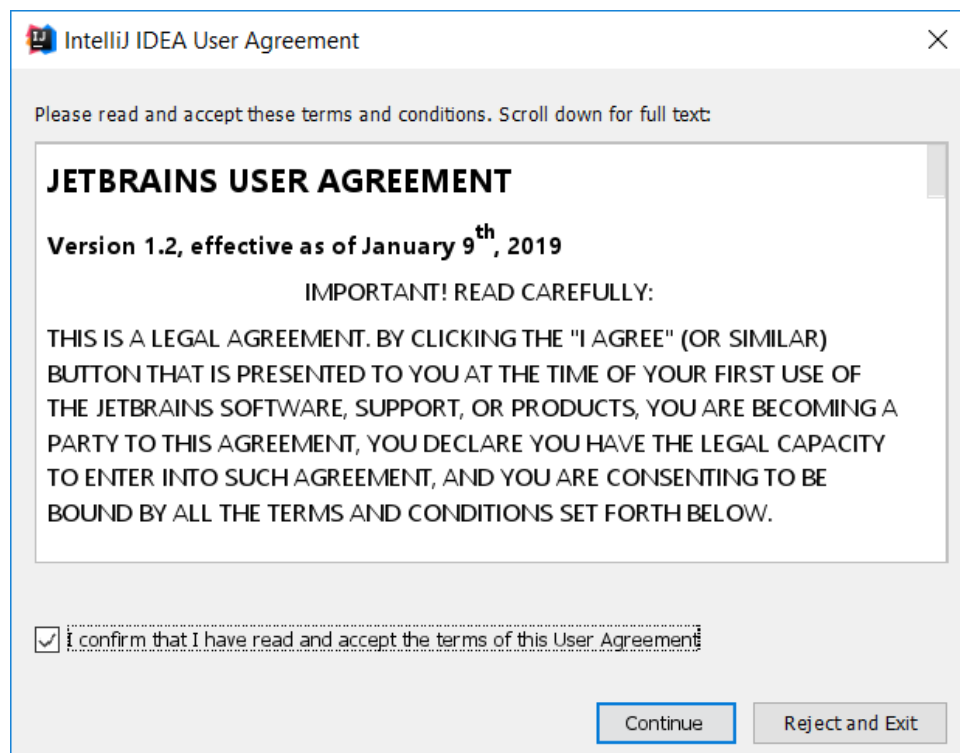


Рисунок 7 – Лицензионное соглашение

На рисунке 8 нажимаем *Don't send*, что позволит не отправлять данные и статистику использования *IntelliJ IDEA*.

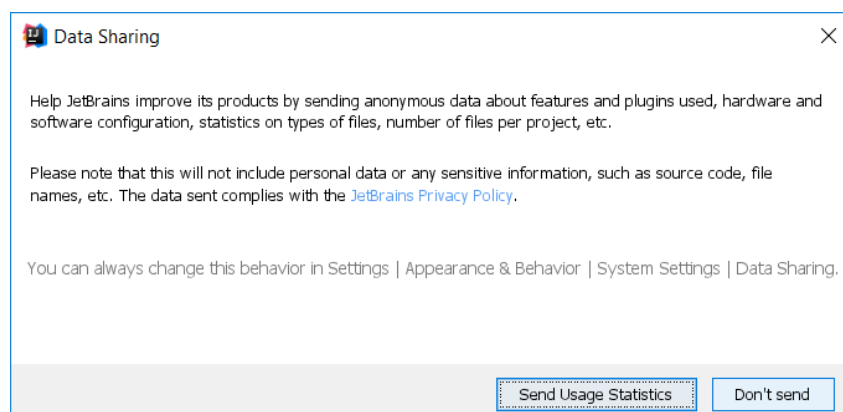


Рисунок 8 – Сбор статистики

Выбираем цветовую схему и нажимаем *Skip Remaining and Set Defaults* (рисунок 9).

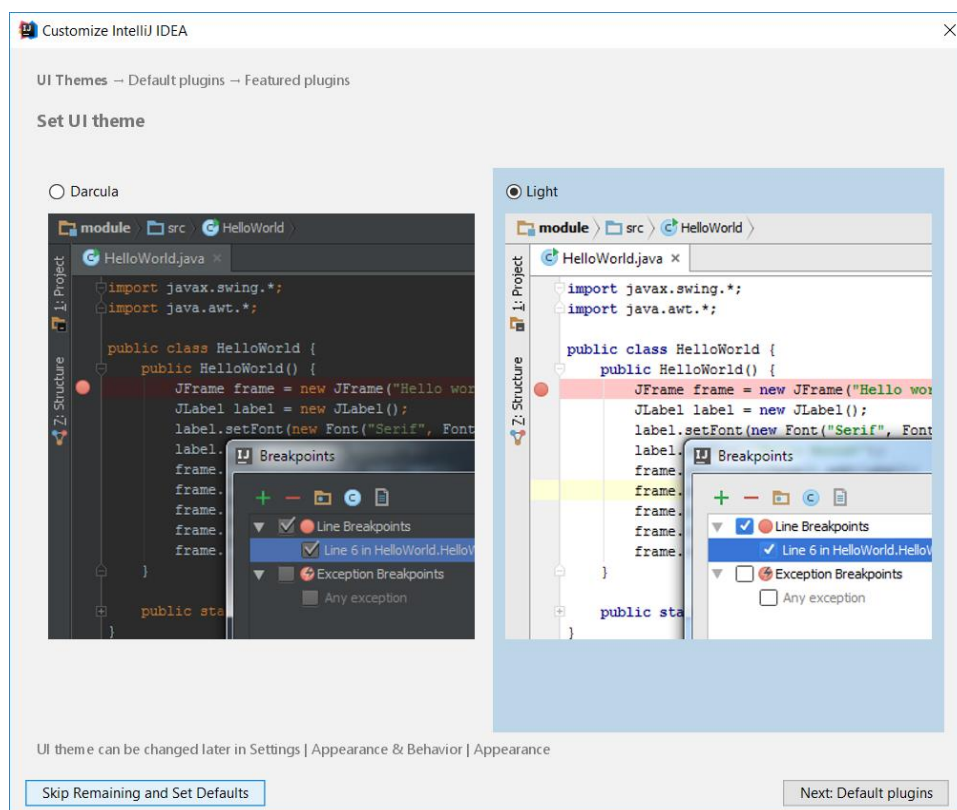


Рисунок 9 – Выбор цветовой схемы

Открылось окно активации. Вводим аккаунт от *JetBrains* (рисунок 10). Также можно просто ввести код активации (его можно найти в личном кабинете на сайте *JetBrains*).

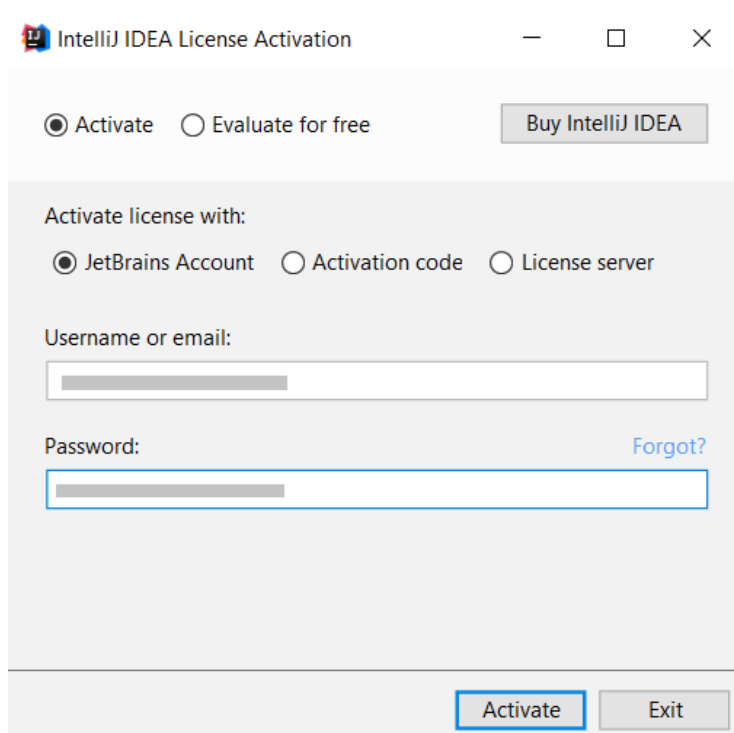


Рисунок 10 – Активация

Появляется окно, где нажимаем *Create New Project* (рисунок 11).

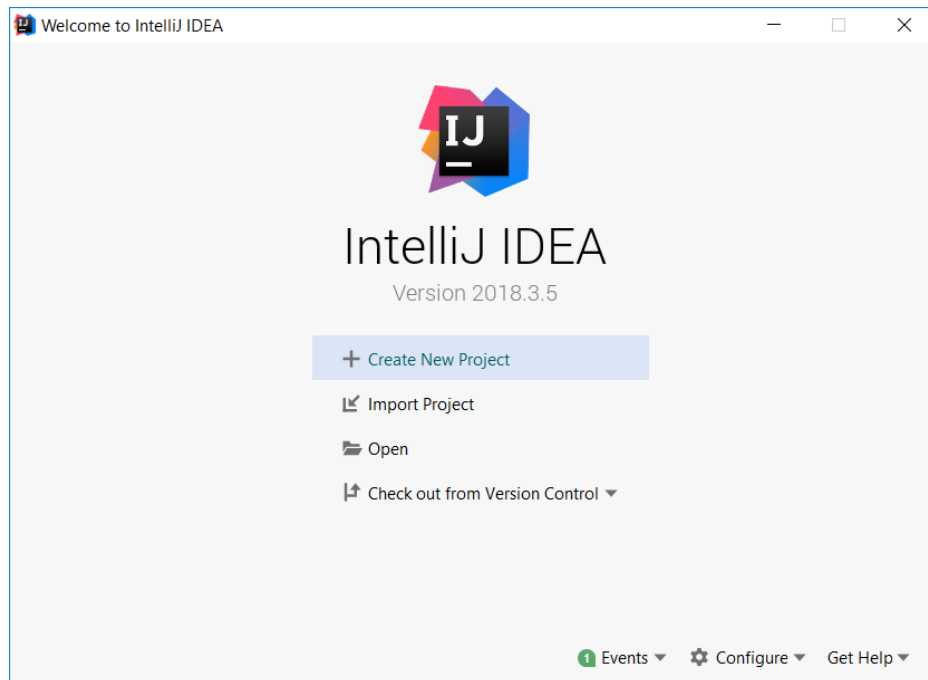


Рисунок 11 – Создание проекта

Перед созданием проекта нам нужно скачать контейнер сервлетов *Apache Tomcat* (иначе говоря – веб-сервер). Это можно сделать по ссылке: <https://tomcat.apache.org/download-90.cgi>

Скачиваем архив в зависимости от операционной системы (рисунок 12).

- Core:
 - [zip](#) ([pgp](#), [sha512](#))
 - [tar.gz](#) ([pgp](#), [sha512](#))
 - [32-bit Windows zip](#) ([pgp](#), [sha512](#))
 - [64-bit Windows zip](#) ([pgp](#), [sha512](#))

Рисунок 12 – Загрузчики Apache Tomcat

Распаковываем данный архив в любое место. В итоге получится примерно так: *C:\apache-tomcat-9.0.13*.

Возвращаемся к созданию проекта. Выбираем тип проекта – *Java Enterprise* (также ставим галочку напротив *Web Application*). В поле *Project SDK* выбираем установленный *JDK*, а в поле *Application Server* – *Apache Tomcat*, который мы скачали выше. Нажимаем *Next* (рисунок 13). Если у вас нет пункта *Java Enterprise*, то, скорее всего, вы установили *Community* версию, а не *Ultimate*.

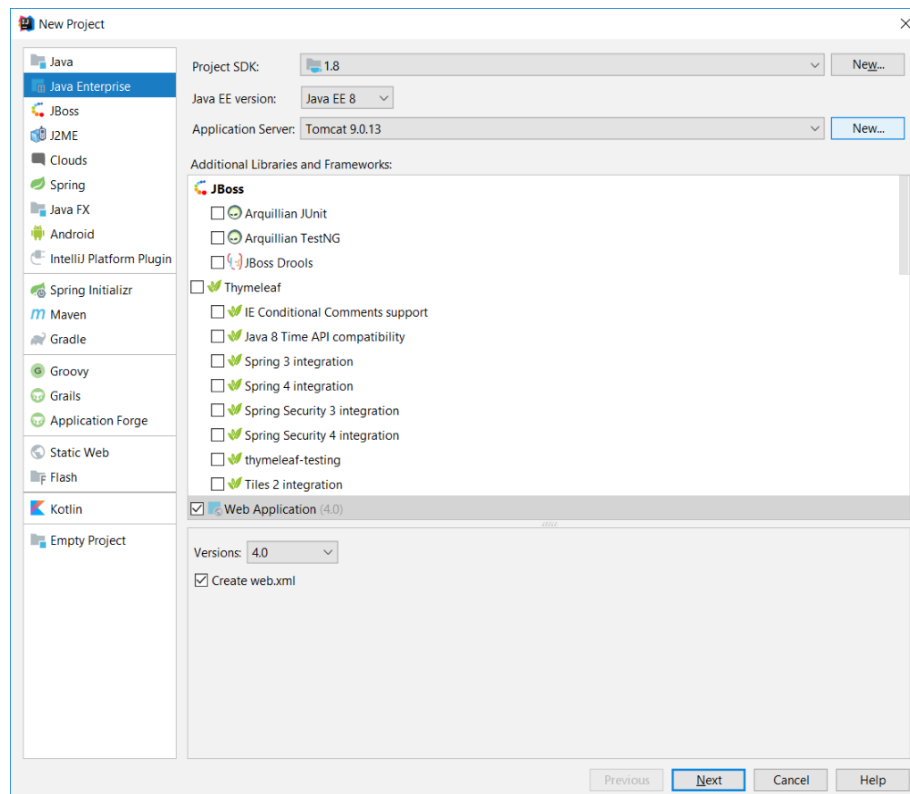


Рисунок 13 – Создание проекта

В поле *Project name* вводим имя нашего проекта. Создаем каталог (в данном случае *Checkboxes*) и указываем к нему путь. Нажимаем *Finish* (рисунок 14).

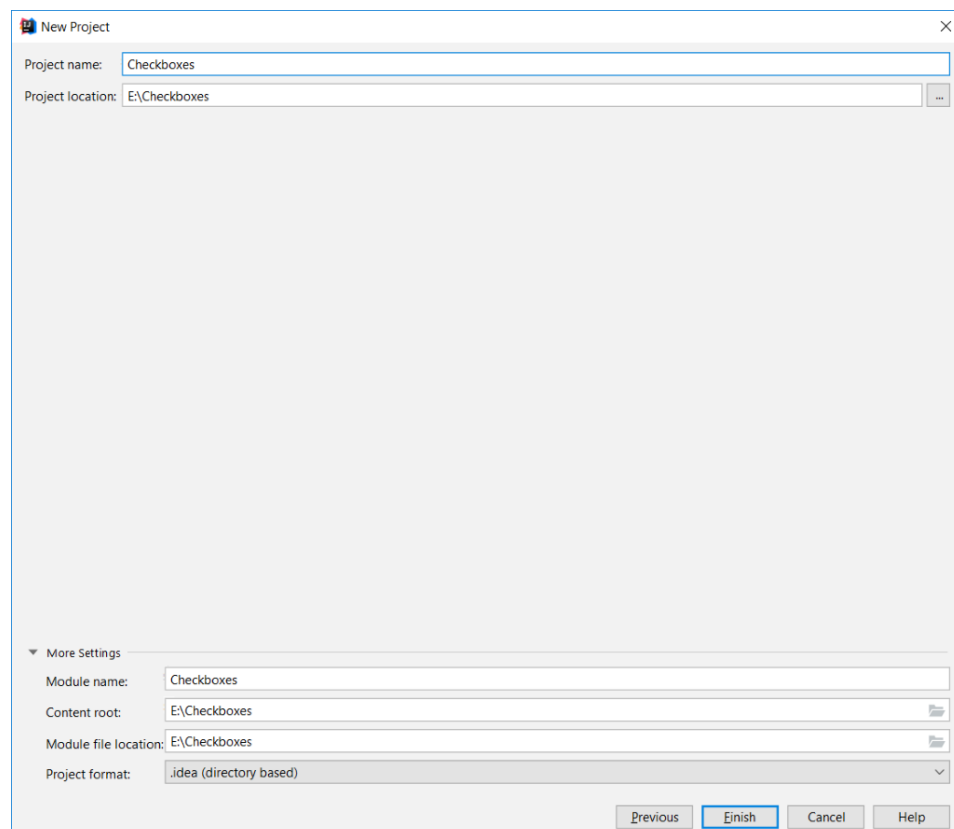


Рисунок 14 – Именованние проекта

Итак, у нас появилась страница, как на рисунке 15.

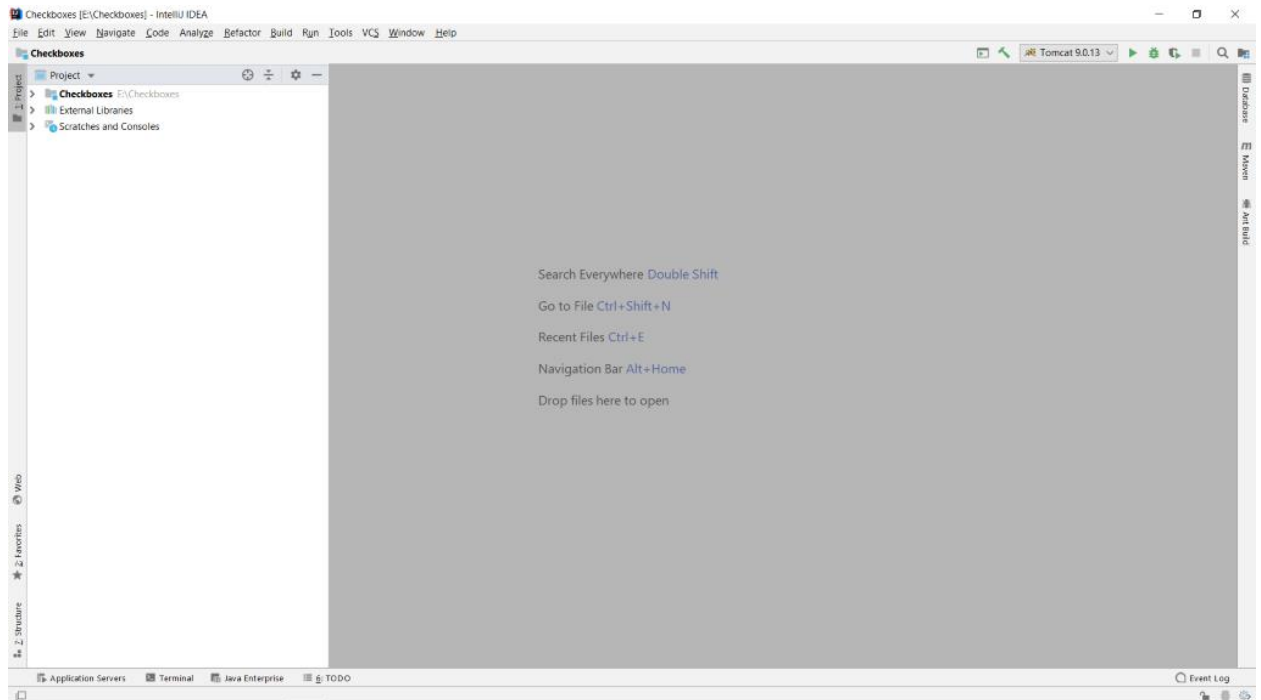


Рисунок 15 – Стартовая страница

Рассмотрим простую задачу для демонстрации запуска *JSP-страницы* (рисунок 16).

Лучшие 20 Обсудить задачу в форуме (-): KorotchiKov Nikita\JSP\1 - "Проверить чекбоксы" 219083 Коротчиков Никита, ИТП-31, февраль 2019

Баллов: 10
Решений: 0
Сложности: 1

Условие

Дана html-страница (см. пример). На ней находится список предметов. Пользователь, заходя на страницу, должен выбрать список предметов (может и не выбрать), которым он обучается, а затем отправить запрос на сервер. Итак, ваша задача состоит в том, чтобы определить (вывести) список предметов (значения которых содержатся в атрибуте value тега input), которые отметил пользователь. Каждый предмет, который будет выведен, нужно поместить в тег <p>.

Пример

Например, дана html-страница:

Checkboxes

localhost:8080/checkboxes.html

☒ Математика ☐ Физика ☒ Информатика

Отправить

Если пользователь нажмет на кнопку отправить, то он получит следующее (обратите внимание на URL):

localhost:8080/checkboxes.jsp

localhost:8080/checkboxes.jsp

subject1

subject3

Информация	
Расширение файла-решения	.jsp
Метод передачи параметров	POST
Атрибуты URL-запроса	subject
Подсказки	Просмотр
Решение	Просмотр

Отправка решения:

Файл не выбран

Выберите файл

Отправить

Показать редактор

DATA

Рисунок 16 – Задача

Обратите внимание, что решение данной задачи – *JSP-страница* (.jsp-файл). Поэтому создадим ее (также обратите внимание на путь создания – каталог web), как показано на рисунке 17.

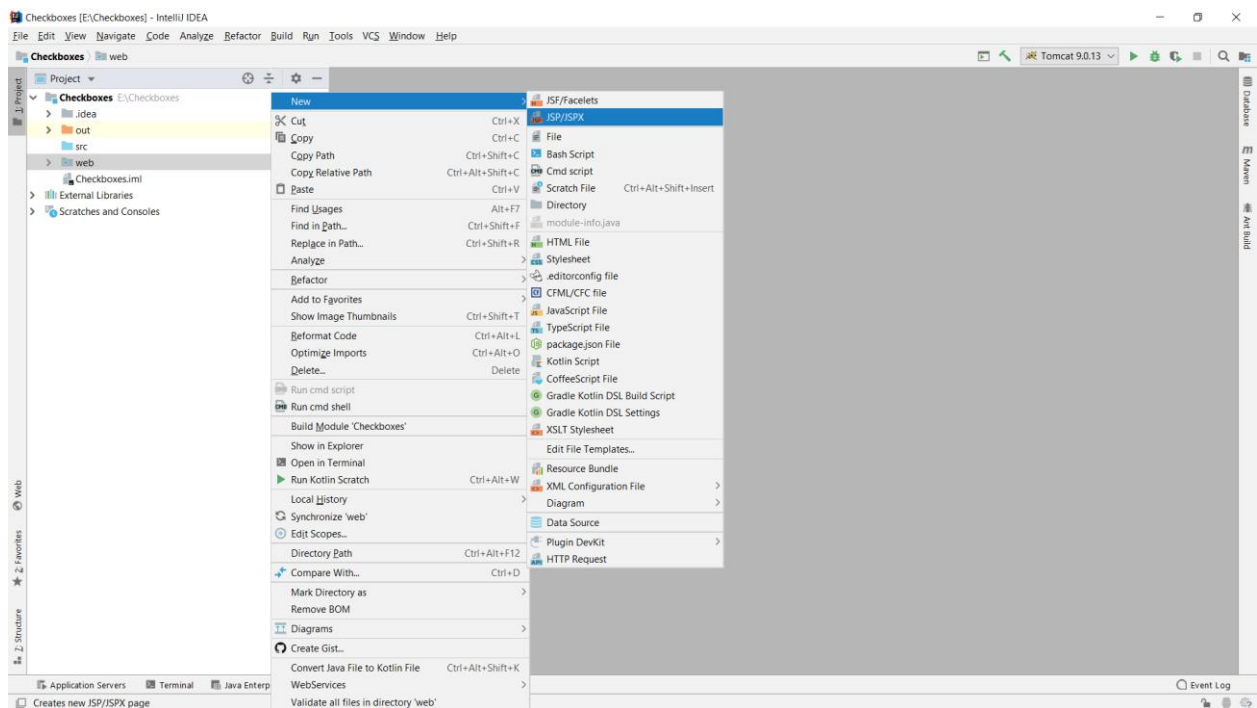


Рисунок 17 – Создание JSP-страницы

Вводим название нашей *JSP-страницы* и нажимаем *OK* (рисунок 18).

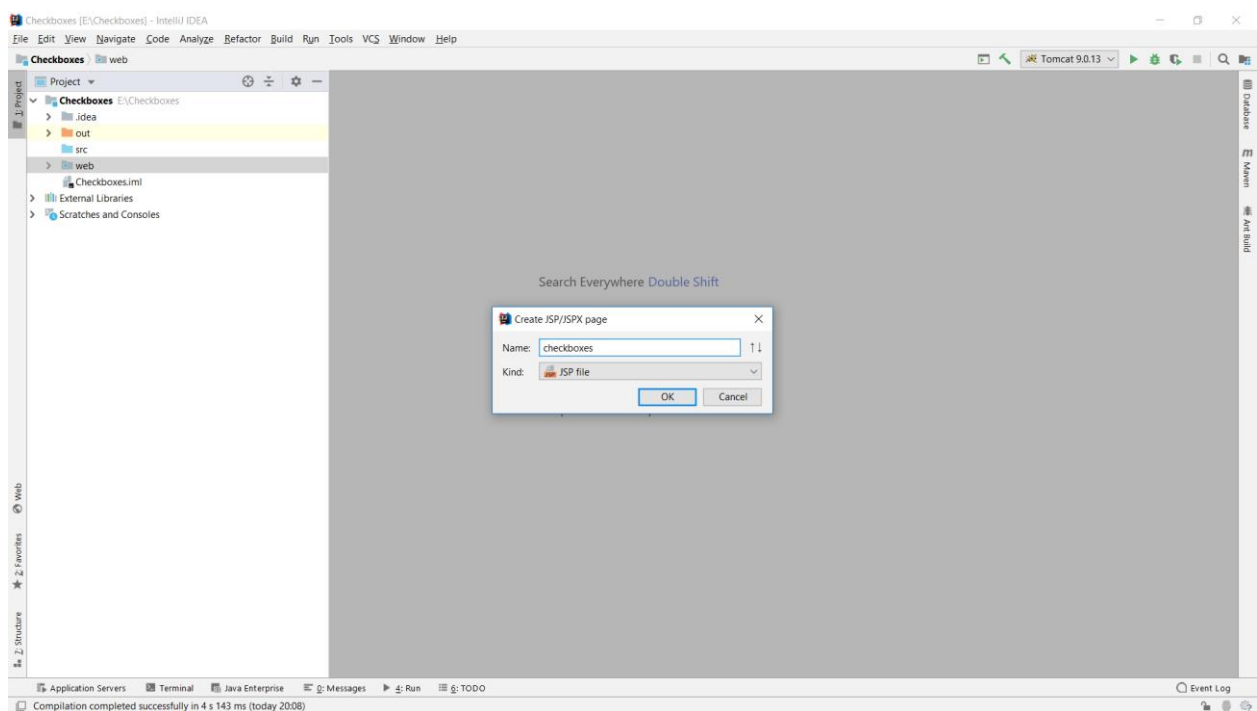


Рисунок 18 – Создание JSP-страницы

Итак, созданная *JSP-страница* показан на рисунке 19.

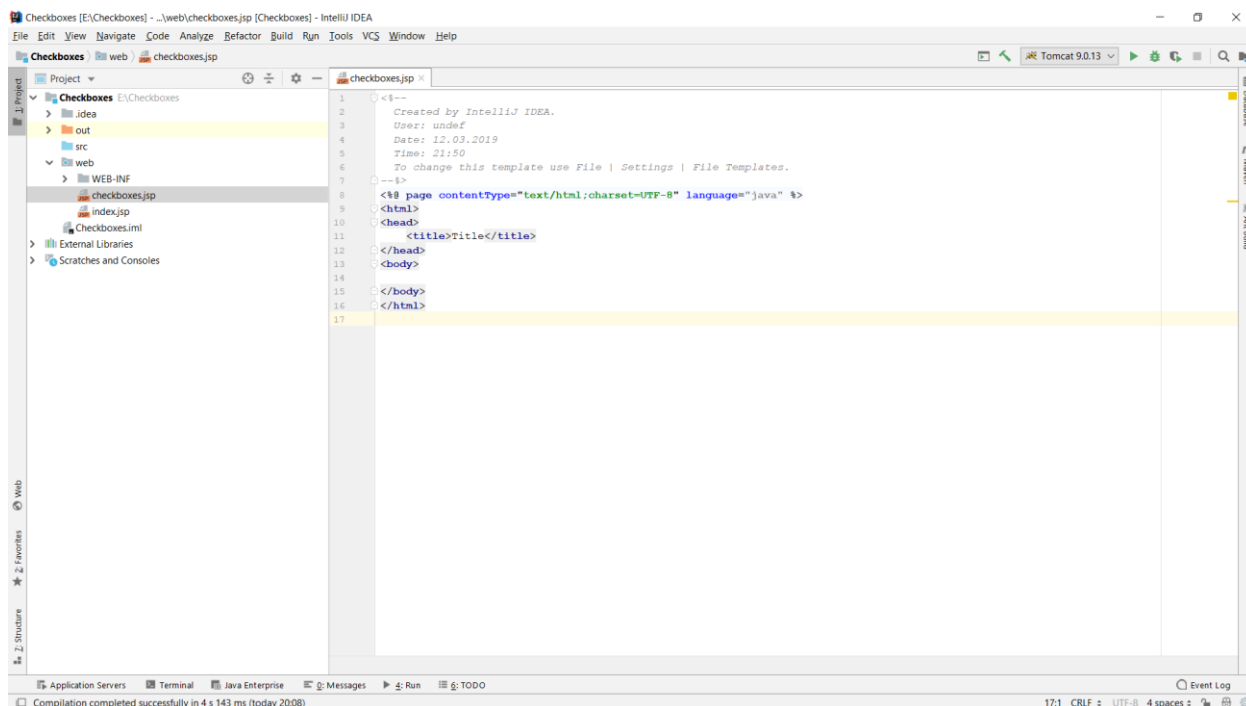


Рисунок 19 – Созданная JSP-страница

Так как это обучение по установке и запуску, то мы не будем акцентировать внимание на решении данной задачи. Оно представлено на рисунке 20.

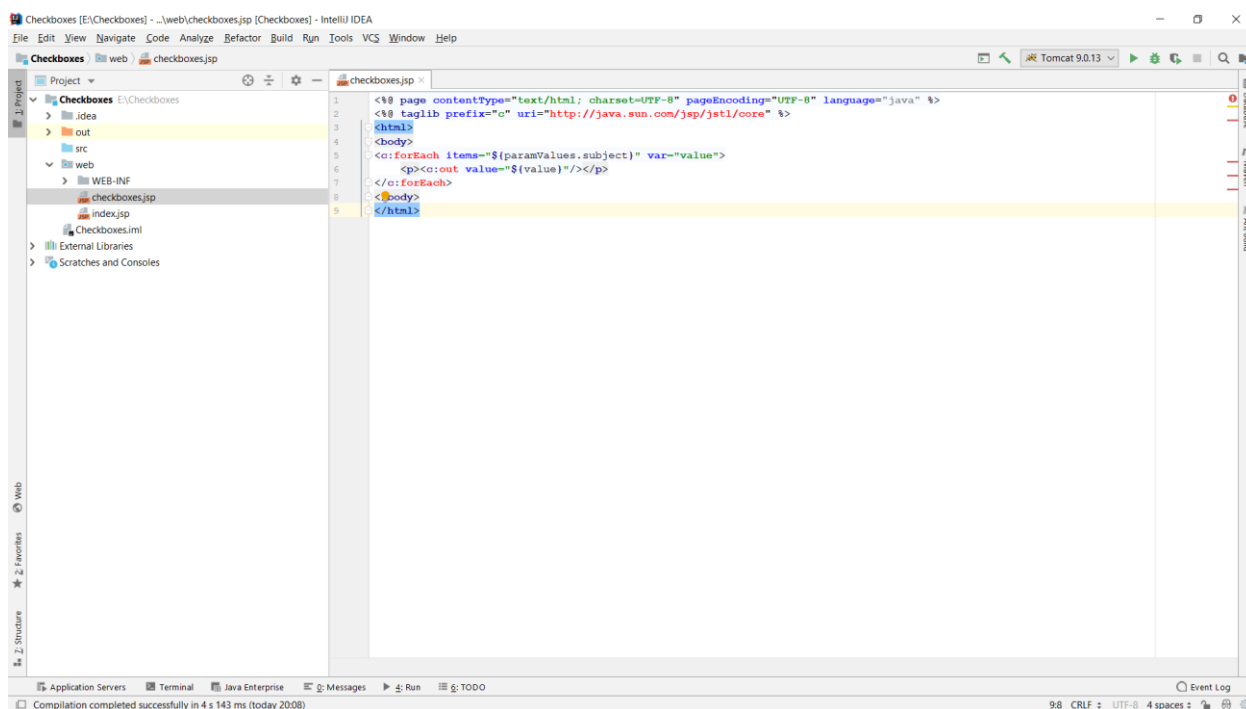


Рисунок 20 – Решение задачи

Обратите внимание на то, что у нас появились ошибки. Дело в том, что не подключена библиотека тегов *JSTL*. Нам нужны две *.jar* библиотеки: библиотека самих тегов и их реализация, они доступны по следующим ссылкам соответственно:

<https://mvnrepository.com/artifact/javax.servlet.jsp.jstl/javax.servlet.jsp.jstl-api/1.2.1>

<https://mvnrepository.com/artifact/org.glassfish.web/javax.servlet.jsp.jstl/1.2.1>

Скачиваем данные библиотеки (вообще говоря, лучшее решение – это создать каталог *lib* в *WEB-INF*, куда и поместить данные библиотеки). Остается импортировать данные библиотеки. Как это делается показано на рисунке 21.

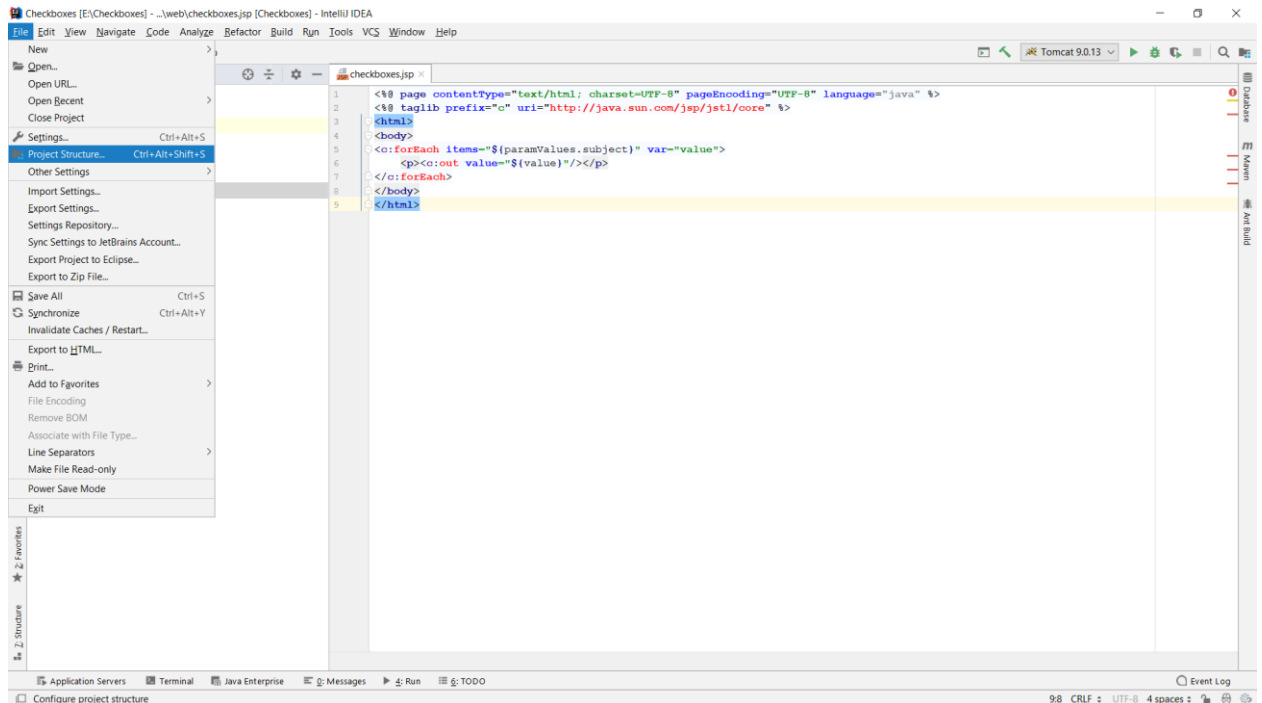


Рисунок 21 – Импорт библиотек

Заходим во вкладку *Libraries*, нажимаем на знак «+» и выбираем *Java* (рисунок 22).

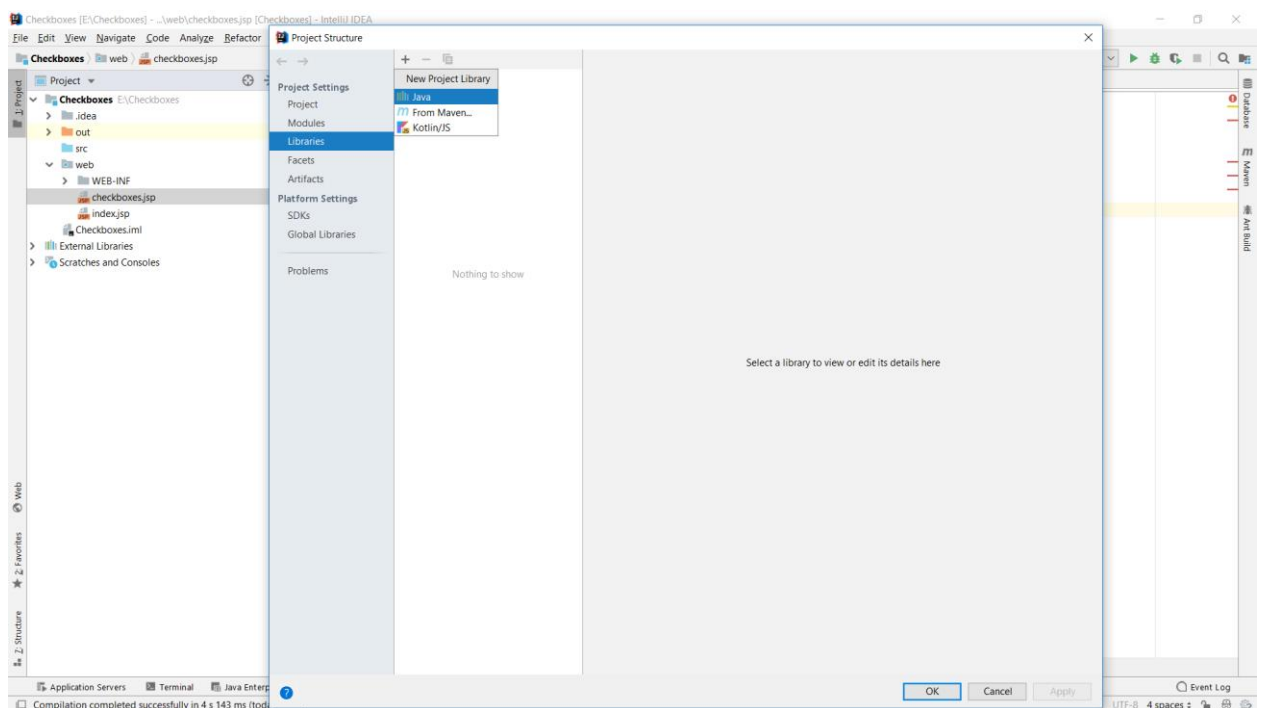


Рисунок 22 – Импорт библиотек

Появится окно, где нужно выбрать скачанные библиотеки. На рисунке 23 показан результат данных действий. Нажимаем **OK**.

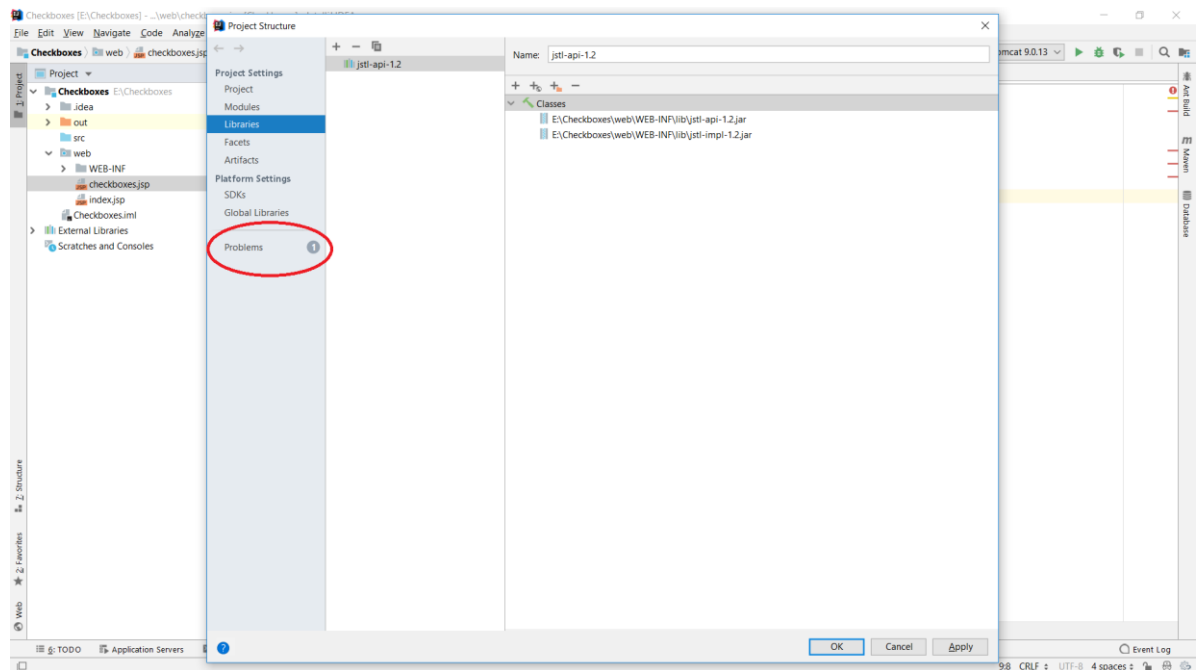


Рисунок 23 – Импорт библиотек

Видим, что появилась проблема. Для ее разрешения переходим во вкладку *Artifacts*, нажимаем *Fix* и добавляем библиотеку в артефакт, как показано на рисунке 24. Нажимаем **OK**. Проблема должна исчезнуть.

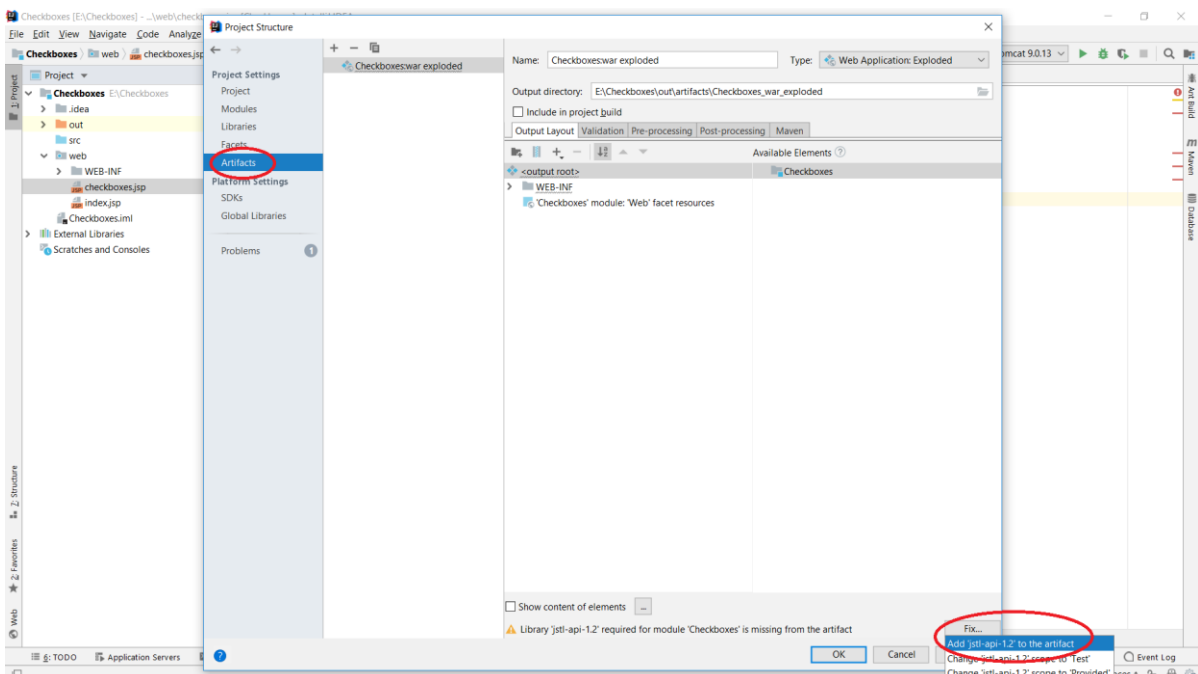


Рисунок 24 – Решение проблемы

На рисунке 25 показано, что ошибки, которые возникали раньше, больше не появляются.

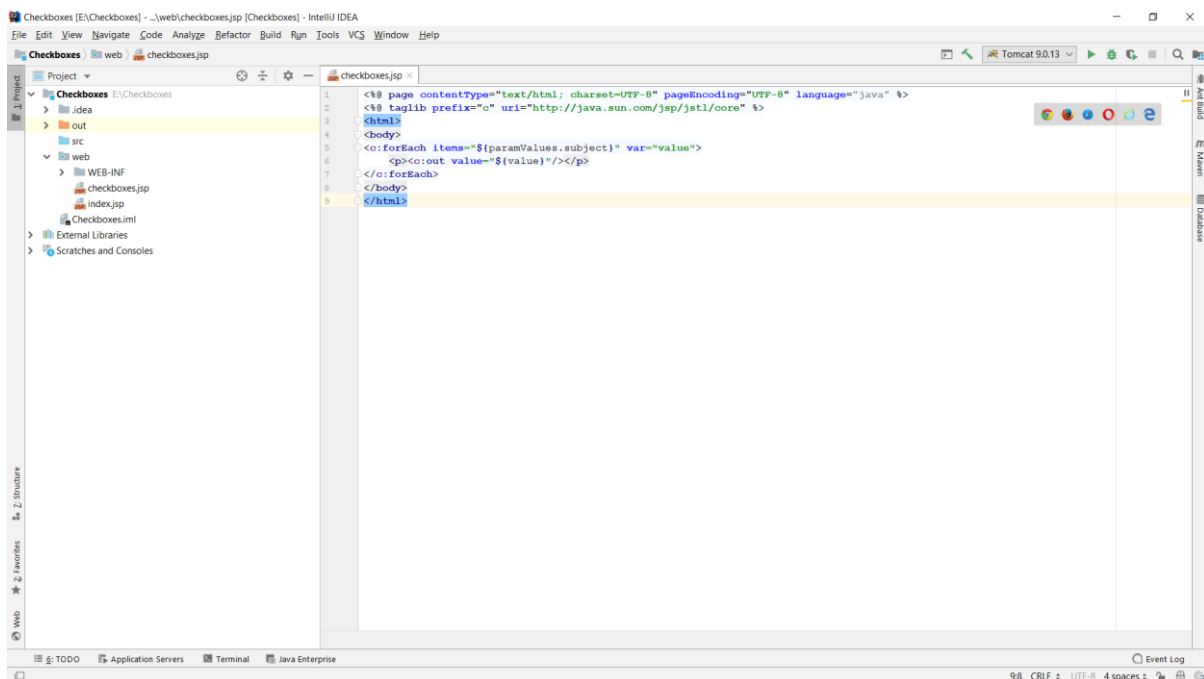


Рисунок 25 – Корректная JSP-страница

Также в задаче сказано, что имеется и исходный *HTML-файл*, где пользователь будет выбирать предметы и нажимать на кнопку отправить. Скачиваем его и помещаем в наш проект – каталог *web* (рисунок 26). Также важно заметить, что у тега *<form>* определены два атрибута: *action* и *method*. Атрибут *action* определяет адрес (в данном случае *JSP-страница index.jsp*), куда будет отправлена форма, а атрибут *method* определяет метод передачи параметров запроса.

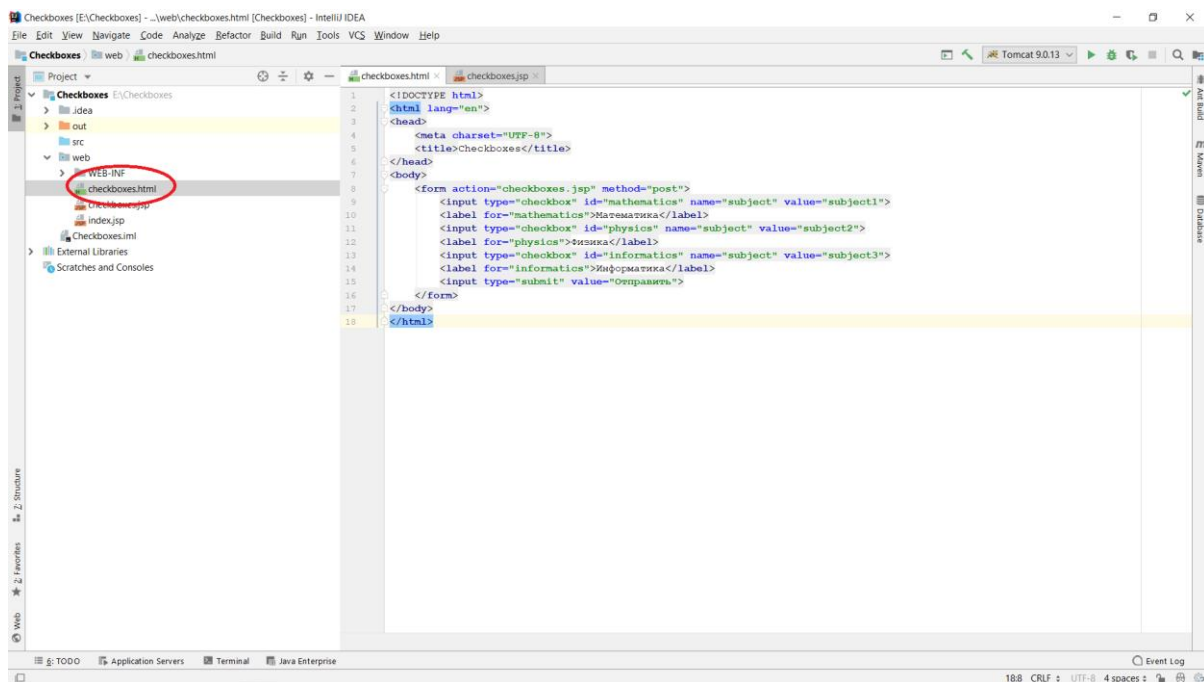


Рисунок 26 – Добавление HTML-файла

Запускаем проект (рисунок 27).

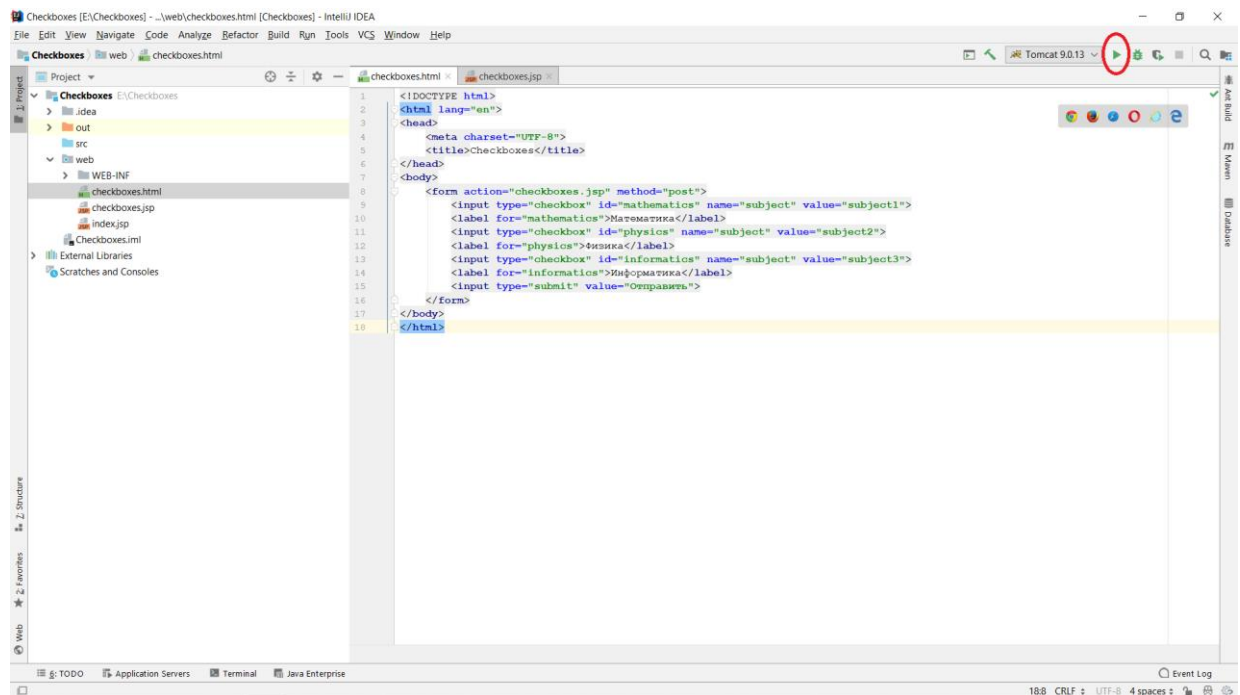


Рисунок 27 – Запуск проекта

После нескольких секунд ожидания автоматически откроется браузер. Вероятно, вы увидите запись `$END$`. Это открывается страница `index.jsp` при запуске, поэтому введите `URL` самостоятельно, как показано на рисунке 28. Результаты работы *HTML-страницы* и *JSP-страницы* показаны на рисунках 28 и 29 соответственно.

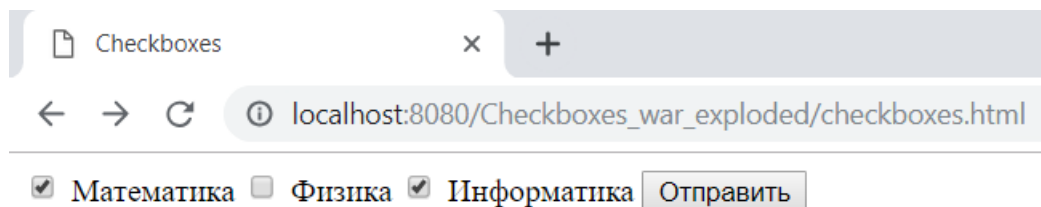


Рисунок 28 – Результат работы HTML-страницы

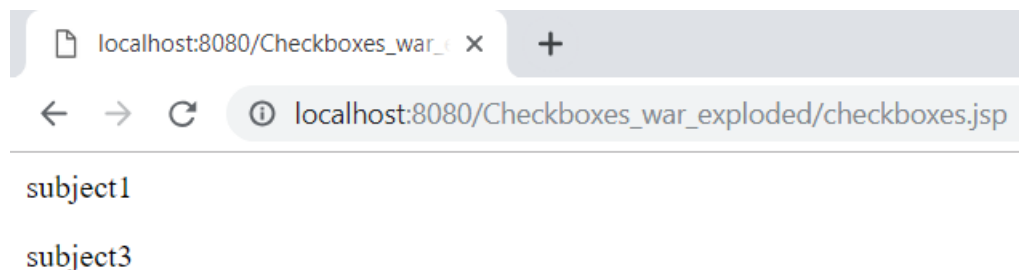


Рисунок 29 – Результат работы JSP-страницы

Если нужно избавиться от имени артефакта в строке `URL`, то нужно проделать следующие действия (в данном случае артефакт имеет имя `Checkboxes_war_exploded`). Нажимаем *Edit Configurations...*, как показано на рисунке 30.

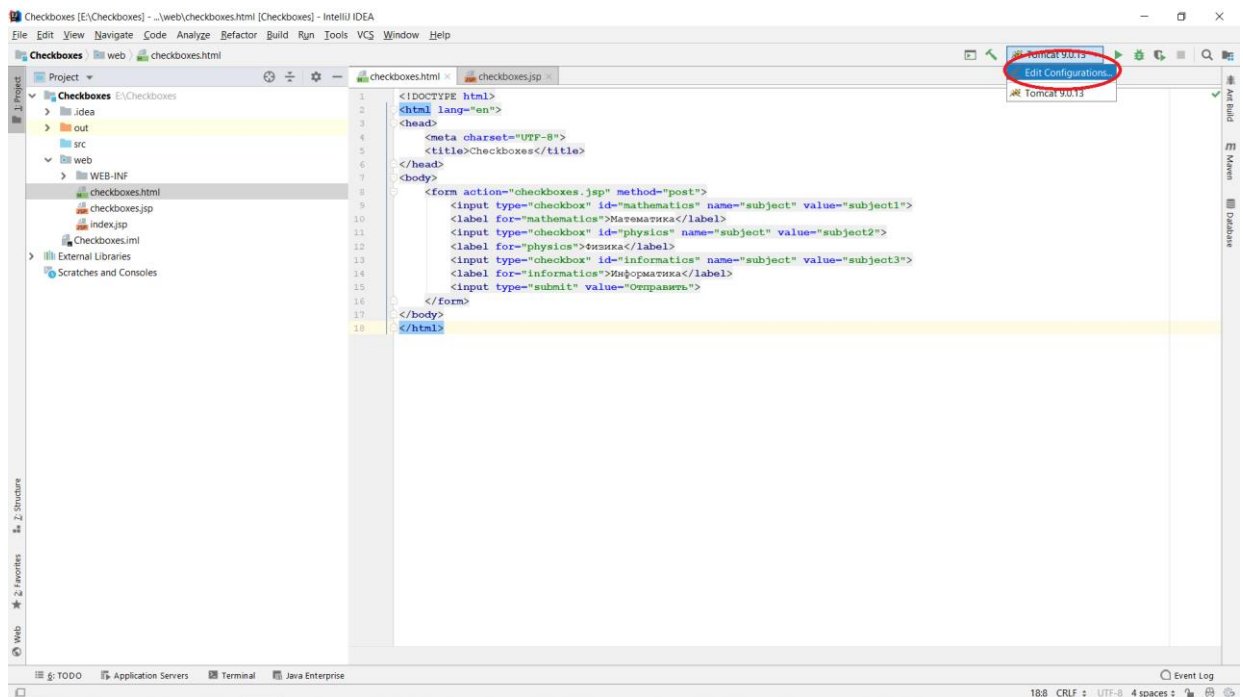


Рисунок 30 – Настройка конфигураций

Переходим во вкладку *Deployment*, где в поле *Application context* выбираем пустую строку. Нажимаем *OK* (рисунок 31).

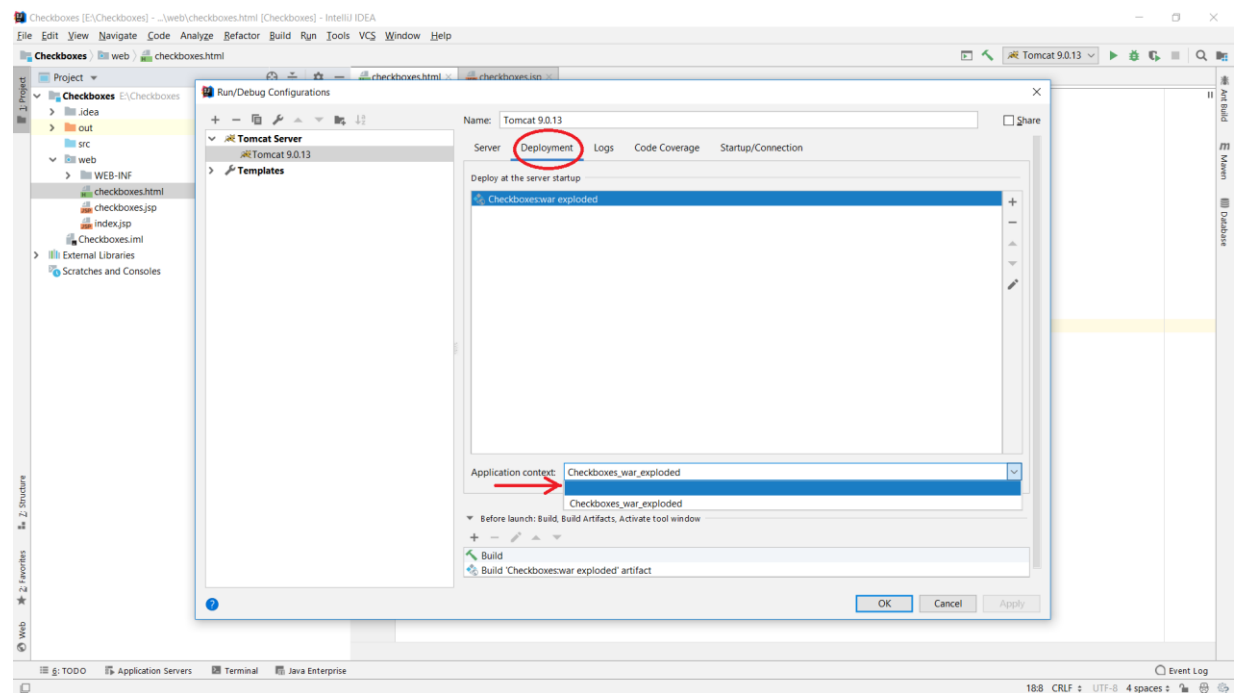


Рисунок 31 – Выбор контекста

После привычного запуска и ожидания получим результаты, которые представлены на рисунках 32 и 33 (обратите внимание на *URL*, имя артефакта уже не указывается).

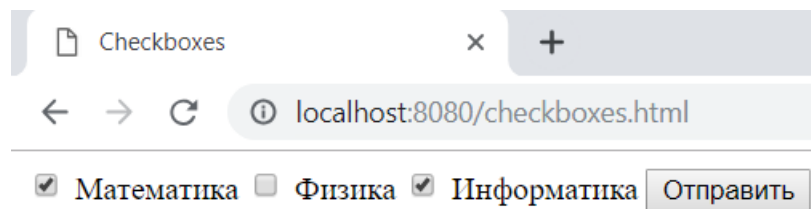


Рисунок 32 – Результат работы HTML-страницы

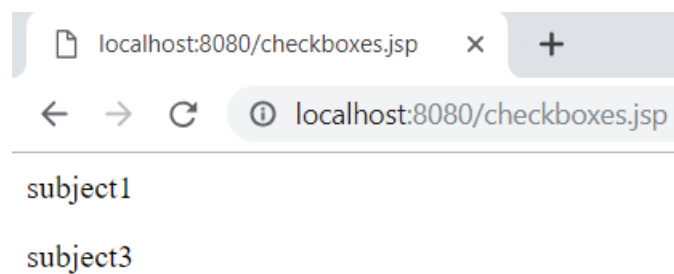


Рисунок 33 – Результат работы JSP-страницы