

Аннотация: этот документ является практическим руководством к Coding Bootcamp.

Week 0 Day 0.5

**Академия Ковалевского**

**Содержание**

[**1. Тестирование кода**](#_wp6y9vkzi41v) **2**

[Тестирование](#_mhrmlb8qgt10) 2

[JUnit](#_di2fduspcplm) 2

[Zeus](#_84eu752sdxks) 2

[Checkstyle](#_lliv2ahbz2b2) 3

[Рекомендации](#_2hvvxqfxbijy) 3

[Задания](#_o168il22p206) 3

[**2. Система контроля версий**](#_jjtk0oolyahq) **4**

[Контроль версий](#_jer3wsvx1qts) 4

[Git](#_m5xuul3f1wqs) 4

[Github](#_poe1lgesyf1g) 4

[Рекомендации](#_i1u0uihws0j2) 5

[Задания](#_5svta4oh5tr7) 5

# 

# 1. Тестирование кода

## Тестирование

**Тестирование** — это процесс проверки функционала программы с целью подтверждения того, что она работает в соответствии с определенными требованиями. Это крайне важный этап разработки ПО, который помогает создавать качественный продукт.

Тестирование, которое пишется, непосредственно, на уровне разработчика называется Unit-тестирование. Оно делится на две большие группы — ручное и автоматизированное тестирование.

## JUnit

**JUnit** — это фреймворк, разработанный для автоматического тестирования программ, написанных с использованием технологии Java.

**JUnit тест** — это простой класс Java, который программист пишет для тестирования кода. Как правило, один метод отвечает за тестирование одного кейса.

## Zeus

**Zeus** — это программа на Java, написанная нами специально для тестирования кода учеников CodingBootcamp при помощи JUnit. Она компилируется, запускается и выполняется аналогично другим Java-программам.

## 

## Checkstyle

**Checkstyle** — это инструмент анализа кода, используемый при разработке программного обеспечения для проверки, соответствует ли исходный код Java указанным правилам кодирования.

## Рекомендации

* Посмотреть видеоурок “[Популярные ошибки по Checkstyle](https://www.youtube.com/watch?v=5_aCBSbtADE&ab_channel=%D0%9F%D0%B5%D1%82%D1%80%D0%90%D1%80%D1%81%D0%B5%D0%BD%D1%82%D1%8C%D0%B5%D0%B2)”

## Задания

* Скачать JAR файл с программой Zeus
* [Прочитать](https://github.com/Kovalevskyi-Academy/Zeus) краткую справку о том, как пользоваться Zeus
* Запустить Zeus с вашим проектом и исправить найденные ошибки

# 

# 2. Система контроля версий

## Контроль версий

Система контроля версий (Version Control System, VCS) представляет собой программное обеспечение, которое позволяет отслеживать изменения в документах, при необходимости производить их откат, определять, кто и когда внес исправления и т.п. Основным понятием VCS является репозиторий (repository) – специальное хранилище файлов и папок проекта, изменения в которых отслеживаются.

## Git

Git — распределенная система контроля версий, разработанная Линусом Торвальдсом для работы над ядром операционной системы Linux. Распределенные системы контроля версий (Distributed Version Control System, DVCS) позволяют хранить репозиторий (его копию) у каждого разработчика, работающего с данной системой. При работе с такой системой, пользователи периодически синхронизируют свои локальные репозитории с центральным и работают непосредственно со своей локальной копией.

## Github

GitHub — сервис онлайн-хостинга репозиториев, обладающий всеми функциями распределенного контроля версий и функциональностью управления исходным кодом — всё, что поддерживает Git и даже больше. Обычно он используется вместе с Git и даёт разработчикам возможность сохранять их код онлайн, а затем взаимодействовать с другими разработчиками в разных проектах. В нашем курсе мы будем использовать аналогичный сервис, предоставленный Google — Cloud Source Repositories.

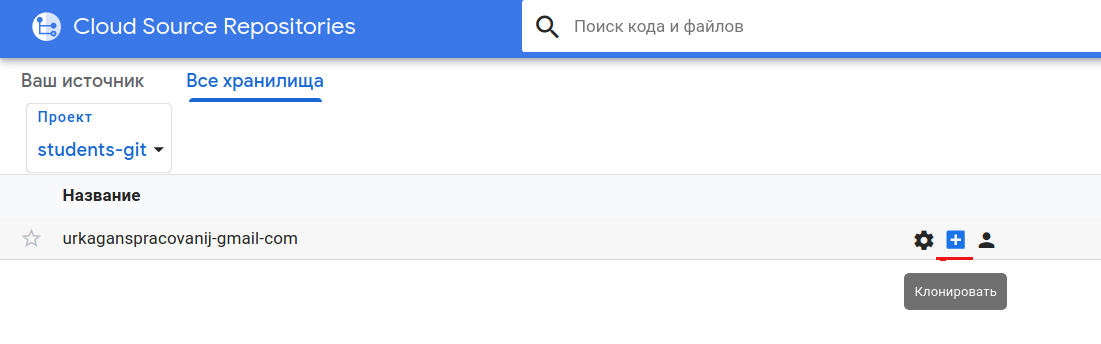
## 

## Рекомендации

* Посмотреть видео урок “[Как использовать систему контроля версий git”](https://www.youtube.com/watch?v=wvqiGJu3YmQ&list=PLsQAG1V_t58Cd5Anx_wJ7-KIzCA8SPOa1&ab_channel=KovalevskyiAcademy)

## Задания

* [Скачайте](https://git-scm.com/) и [установите](https://www.youtube.com/watch?v=N38Pol6Mgfk&ab_channel=JavaScript.ru) Git
* Войдите в [Cloud Source Repositories](https://source.cloud.google.com/students-git) используя свою почту
* Выберите доступное вам хранилище и клонируйте его

**

*Рекомендуем использовать инструкцию “Аутентификация по SSH (SSH authentication)”, но можно и через “Google Cloud SDK”.*

***Учтите****: скорее всего при подключении через SSH нужно будет воспользоваться командой ssh-add*

***Учтите****: рекомендуем не использовать в этом задание PuTTY на Windows*

* Сделайте commit и push всего вашего когда, который был сделан за прошедшие дни на удаленный репозиторий