

Лабораторная работа 1: Использование IDE

- 1) Скачать и установить Visual Studio Code
- 2) Сконфигурировать для запуска и отладки кода (любой из языков java, python, c#).
 - a. Продемонстрировать на примере, что код компилируется, работает.
- 3) Установить шрифт FiraCode
- 4) Написать простой проект (как пример: чтение файла, консольный калькулятор)
 - a. Наличие не менее 3 классов
 - b. Наличие порядка 10 функций внутри классов
- 5) Написать тесты к проекту.
 - a. Проверка входных данных
 - b. Проверка результата
 - c. Тесты должны быть распознаны IDE
 - i. Пример для Python - <https://code.visualstudio.com/docs/python/testing>
 - ii. Пример для Java - <https://code.visualstudio.com/docs/java/java-testing>
- 6) Активировать Visual Studio Code CLI:
 - a. Уметь отобразить разницу между 2 файлами
- 7) Написать 5 собственных Снимков. (Собственные аналоги if, for, while – обязательное наличие переменных для замещения)
- 8) Изучить и описать наиболее распространённые HotKeys – не менее 10 штук.
- 9) Задokumentировать код используя расширения – генераторы документации.
 - a. <https://marketplace.visualstudio.com/items?itemName=KeeganBruer.java-doc-generator>
 - b. <https://marketplace.visualstudio.com/items?itemName=cschlosser.doxdocgen>
 - c. <https://marketplace.visualstudio.com/items?itemName=njpwerner.autodocstring>
 - d. или аналоги.
- 10) Установить <https://marketplace.visualstudio.com/items?itemName=MS-vsliveshare.vsliveshare> и мочь продемонстрировать свою работу с помощью данного расширения. (преподавателю или одногруппнику)

Отчет – Описание этапов выполнения заданий лабораторной работы со скриншотами, исходники проекта.

Примечание: При огромном желании возможна замена IDE на другое по согласованию с преподавателем. Однако главная цель лабораторной работы: использовать максимум возможностей IDE, поэтому приветствуется установка расширений и конфигурация под пользователя.

Оценка: 10 Баллов всего. 1 пункт = 1 балл.

Лабораторная работа 2: Изучение системы управления версиями Git

Каждому студенту:

- a. Каждому студенту завести аккаунт на GitHub (давно пора!)
- b. Для проекта из первой лабораторной работы создать git репозиторий.
- c. Создание веток по модели GitFlow -
 - i. Обязательное наличие веток: фич, релиз, девелоп, хот фикс.
- d. Дополнить функциональность своих приложений (не менее 5 полноценных фич – считывание нового формата данных, расчет новой математической функции) с помощью новых классов и функций. Каждую новую функциональность необходимо оформлять как фичу. Каждые 2 фичи релиз!
- e. В репозитории обязательно оформлен ReadMe.
- f. Наличие тегов.
- g. Submodules – Создание хотя бы двух подмодулей с исходниками. Уметь продемонстрировать работу с ними – push, pull ...
- h. LFS организовать заливку файла размером более 100 Мб.

Разбиваемся по парам:

- 1) Первый студент заводит репозиторий (небольшой проект, можно взять из 1 лабораторной работы), второй делает в нее Pull request (добавить новый класс, изменить/дополнить функцию первого студента).
- 2) Поменяться местами)

Также всем студентам обязательно подготовить справочник по основным командам Git с примерами. Уметь кратко ответить на вопросы о предназначении основных команд Git.

Почитать:

https://danielkummer.github.io/git-flow-cheatsheet/index.ru_RU.html

<https://git-scm.com/book/ru/v1>

<https://www.atlassian.com/ru/git>

Отчет – Описание этапов выполнения заданий лабораторной работы + справочник команд GIT

Примечание: Возможно использование Gitlab, Bitbucket в качестве аналога Github. Также очень сильно приветствуется (почти требуется) разработка в VS Code и использование в нем расширения для работы с Git.

Оценка: 15 Баллов всего. 10 баллов – все выполненные пункты лабораторной. 5 баллов за 5 рандомных вопросов по командам git.