Лабораторная работа 1: Настройка IDE
Нужно изучить и описать выбранную интегрированную среду разработки в отчете в объёме не менее десяти страниц

1) Скачать и установить Visual Studio Code
2) Сконфигурировать для запуска и отладки кода (любой из 3 языков java, python, c++).
3) Написать простой проект (чтение файла, простой консольный калькулятор)
4) Написать тесты к проекту.
5) Изучить Visual Studio Code CLI:
   описать не менее 10 команд используемых в данном интерфейсе.
   Написать 5 собственных Сниппетов
6) Изучить и описать наиболее распространённые HotKeys
7) Задокументировать код с помощью расширения https://marketplace.visualstudio.com/items?itemName=cschlosser.doxdocgen
8) Установить шрифт FiraCode
9) Установить https://marketplace.visualstudio.com/items?itemName=MS-vsliveshare.vsliveshare и показать чего получилось)
10) Отчет – Пошаговое руководство настройки проекта со скриншотами, исходники проекта.

При огромном желании возможна замена IDE на другое по согласованию с преподавателем.

Лабораторная работа 2: Изучение системы управления версиями Git
Нужно изучить и описать систему управления версиями Git в объёме не более пяти страниц, не считая титульника. Формат сдачи следующий: я задаю вопросы по основным командам Git, вам нужно дать ответы, опираясь на ваши отчеты.
За каждый неправильный ответ отнимается один балл от числа максимального количества баллов за работу.

Разбиваемся по парам и создаем

1) Первый студент заводит репозиторий, второй делает в нее Pull request.
2) Каждому:
   a. Создание веток по модели GitFlow - https://danielkummer.github.io/git-flow-cheatsheet/index.ru_RU.html - Обязательное наличие веток фич, релиз, девелоп, хот фикс.
   b. В репозитории обязательно оформлен ReadMe.
   c. Наличие тегов.
   d. Submodules
   e. LFS
3) Также всем студентам обязательно подготовить справочник по основным командам Git с примерами. Уметь кратко ответить на вопросы о предназначении основных команд Git.

Почитать:

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

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

Лабораторная работа 3: Системы управления репозиториями

1) Нужно изучить, описать и сравнить 2-3 системы управления репозиториями (Github, Gitlab, Bitbucket). В сравнении отобразить разницу – в предоставляемых инструментах, фичах, ограничениях и совместимости.

2) За основу берется лабораторная работа 2 и и оформляется на любой из систем.

3) Также обязательно установить графический интерфейс для Git (GitGui, GitKraken …). Продемонстрировать взаимодействие локального репозитория и системы через графический интерфейс. Показать не менее 10 команд (можно посмотреть из 2 лекции по гиту) в графическом интерфейсе (Куда нажимать, результат).

4) Все пункты из 2 лабораторной работы показать в графическом интерфейсе

5) Для системы управления репозиториями – продемонстрировать навыки

Поиска файла,

Редактирования файла,

Сравнения изменений,

Отслеживания тегов.

В отчете сравнение систем – 5 страниц (отличия и сходства, ключевые особенности) а также результаты выполнения пунктов 2-4.

Лабораторная работа 4: Сравнение систем управления проектами
Нужно изучить, описать и сравнить три выбранные системы управления проектами в объёме не менее десяти страниц.

1) Организовать Scrum доску планирования своего обучения/Работы/Планирования жизни. (каждый студент отдельно) (Trello или аналоги) - - - - Использовать в течении 2 недель минимум!
2) Создать доску небольшого проекта в Trello, либо аналоги. (Предложить свою идею или спросить у преподавателя - в качестве проекта может быть любой предмет по программированию или же проект Разработки плагина для IDE).
    a. Распределиться на команды (2-4) человека и разбиться на роли внутри команды – Аналитик-ВладелецПродукта/Разработчик/Тестировщик. Каждый член команды может совмещать роли, но при этом обязательно отмечать и суметь доказать свою необходимость в команде. Аналитик может подсказывать тестировщику как правильно понимать ТЗ, а тестировщик может спрашивать реализацию у разработчика, чтобы узнать краевые ситуации тестирования.
    b. Обязательная фиксация всех документов, решений, задач и прочего в рамках проекта. Конечная цель такова – каждое поведение кнопки, каждый символ на экране возможно отследить – найти обратный путь от появления его в проде до согласования требования по нему.
    c. Обязательное ведение Git репозитория с прикреплением коммитов в системе управления проектом.
    d. Настройка CI/CD Например – https://docs.gitlab.com/ee/ci/ https://docs.gitlab.com/ee/ci/introduction/index.html#how-gitlab-cicd-works Подразумевается, что даже для лабораторных по другому предмету будет настроен CI/CD по типу – собираться exe фаил, отображаться сайт.
3) Предоставить итеративную версию управления проектом. С количеством релизов – не менее трех (Например 3 лабы по другому предмету, или три фичи).
   Почитать:
    a. https://unito.io/
    b. https://habr.com/ru/company/softmart/blog/316686/

Почитать:

https://worksection.com/blog/it-project-management.html

https://www.bitrix24.ru/features/landing/collaboration/?gclid=Cj0KCQjwsvrpBRCsARIsAKBR_0Litvm2POv7gSQzpVpAiy6-wiRlmInoT1ZD-ZPuGH5XJ7sZjFjMPPAaAsjPEALw_wcB

https://start.worksection.com/?camp=1029458083&gclid=Cj0KCQjwsvrpBRCsARIsAKBR_0I50Oslydtsm3JLUnI1pvBdy1a2OPtr9jpE_KCWoUfBX5JyknXgJw4aAu3zEALw_wcB

https://habr.com/ru/post/173633/

Лабораторная работа 5: Разработка плагинов
Нужно разработать плагин для IDE. По сложности он должен быть сопоставим с редактором Markdown или трекером задач. Если возникают проблемы с идеей для плагина, можете посмотреть существующие и попробовать удачно склонировать имеющиеся, или обсудить идею со мной на занятиях.

В отчете должно быть не менее 20 страниц, это будет что-то типа курсовой работы. В первой половине работы вы должны описать среду, для которой реализуете плагин, кратко описать архитектуру среды и плагинов. Этот блок показывает, насколько вы разобрались с устройством этой среды и её плагинов. Во второй половине работы нужно описать свое решение, можно вставлять куски кода (осторожно! я буду делать code review на весь уведенный мною код в отчете). Здесь вы должны продемонстрировать, что можете разобраться с экосистемой имеющегося программного продукта и внедрить в него свое решение (ваши консольные приложения на плюсах никому не нужны).
Также можно разработать плагин или расширение для других сред (системы управления проектами, системы управления репозиториями и т.д.), но нужно предварительно обсудить задание со мной.
Также возможно выполнение последней лабораторной работы вдвоем, но учтите, что объем работ вырастет минимум в два раза.