

## Лабораторная работа: Настройка системы контроля версий для управления импортом файлов в репозиторий

### 1. Цель работы:

- Изучить механизмы настройки системы контроля версий для управления включением и исключением файлов и каталогов из репозитория.
- Получить практические навыки применения файлов конфигурации игнорирования (например, `.gitignore`) для управления импортируемыми файлами.
- Понять влияние настроек контроля версий на процесс разработки и совместной работы над проектом.

### 2. Задачи:

- [Индивидуально:] Определить специфические требования к импорту файлов в репозиторий для конкретной архитектуры проекта и варианта задания (см. ниже).
- Изучить возможности системы контроля версий (СКВ) для фильтрации файлов на основе их типов, путей и других критериев (например, Git, Mercurial, Subversion).
- Создать и настроить файл конфигурации игнорирования (например, `.gitignore` для Git) в соответствии с определенными требованиями.
- [Индивидуально:] Протестировать настройки, создав тестовые файлы и каталоги, соответствующие архитектуре проекта.
- Зафиксировать изменения в репозитории.
- [Индивидуально:] Проанализировать результаты тестирования и внести необходимые корректировки в файл конфигурации игнорирования.

- Оформить отчет, описывающий процесс выполнения работы, результаты и выводы.

### 3. Инструменты:

- Система контроля версий (рекомендуется Git).
- Текстовый редактор (Visual Studio Code, Sublime Text, Notepad++, и т.д.).
- Командная строка (терминал).

### 4. Варианты заданий (Примеры для адаптации):

- Вариант А (Общий): Игнорировать файлы определенного типа (например, файлы скомпилированного кода, временные файлы, резервные копии). Задача студента: Определить, какие именно типы файлов должны быть проигнорированы, исходя из его архитектуры проекта (например, \*.class для Java, \*.pyc для Python, \*.obj для C++).
- Вариант Б (Пути): Игнорировать файлы и каталоги, расположенные в определенных путях (например, каталоги сборки, каталоги с временными данными). Задача студента: Определить, какие пути должны быть исключены из репозитория, исходя из его структуры проекта (например, build/, tmp/, logs/).
- Вариант В (Специфичные файлы): Включить в репозиторий определенные файлы, находящиеся в каталогах, которые обычно игнорируются (например, файл конфигурации в каталоге с временными данными). Задача студента: Определить, какие файлы должны быть явно включены в репозиторий, несмотря на общие правила игнорирования (используйте ! в .gitignore).
- Вариант Г (Комбинация): Использовать комбинацию правил для игнорирования файлов по типу, пути и другим критериям. Задача студента: Составить комплексное правило игнорирования, учитывающее

несколько факторов (например, игнорировать все файлы `.log` в каталоге `logs/`, кроме файла `logs/important.log`).

- Вариант Д (Расширенный): Использовать продвинутые возможности `.gitignore` или аналогичных файлов (например, игнорирование файлов на основе регулярных выражений, если поддерживается СКВ). Задача студента: Исследовать и применить продвинутые возможности игнорирования файлов, если это применимо к его СКВ и варианту задания. (Например, игнорировать файлы, содержащие определенную строку в имени, но такое потребует скриптов).

- Вариант Е (Связанный с инструментом): Игнорировать файлы, созданные определенным инструментом разработки (IDE, компилятор, отладчик). Задача студента: Определить, какие файлы создает используемый им инструмент и настроить их игнорирование (например, файлы, созданные Visual Studio, IntelliJ IDEA, Eclipse).

5. Порядок выполнения работы (Общий шаблон, адаптируется индивидуально):

1.Подготовительный этап:

- [Индивидуально:] Определение требований: Тщательно проанализируйте архитектуру своего проекта. Определите, какие файлы и каталоги не должны попадать в репозиторий (например, файлы с конфиденциальной информацией, временные файлы, сгенерированный код, файлы, специфичные для вашей среды разработки). Запишите эти требования. Это самый важный шаг!

- [Индивидуально:] Выбор варианта: Выберите один из вариантов заданий (A, B, C, D, E) или сформулируйте свой собственный, исходя из архитектуры проекта и определенных требований.

## 2.Настройка СКВ:

- Инициализируйте репозиторий (если еще не сделано).
- Создайте файл конфигурации игнорирования (например, `.gitignore` для Git).

## 3.Редактирование файла конфигурации:

- [Индивидуально:] Составьте правила игнорирования в файле конфигурации, основываясь на определенных требованиях и выбранном варианте задания. Используйте синтаксис, специфичный для используемой СКВ.

## 4.Тестирование настроек:

- [Индивидуально:] Создайте тестовые файлы и каталоги, которые соответствуют архитектуре вашего проекта и правилам игнорирования. Например, если вы игнорируете все файлы `.log`, создайте несколько таких файлов в разных местах.
- Добавьте и зафиксируйте изменения в репозитории.
- Проверьте статус репозитория (`git status` для Git). Убедитесь, что файлы, которые должны быть проигнорированы, не отображаются в списке неотслеживаемых файлов.

## 5.Анализ и корректировка:

- [Индивидуально:] Проанализируйте результаты тестирования. Если какие-то файлы игнорируются неправильно, внесите корректировки в файл конфигурации и повторите шаги 4 и 5.

Используйте команду `git check-ignore -v <filename>` для отладки (в Git).

#### 6.3. Завершение:

- Зафиксируйте окончательные изменения в репозитории.

#### 6. Содержание отчета:

- Титульный лист (название работы, ФИО, группа, дата).
- Введение:
  - Краткое описание цели работы.
  - [Индивидуально:] Описание архитектуры проекта (ключевые каталоги, типы файлов, используемые инструменты и т.д.).
  - [Индивидуально:] Обоснование выбора варианта задания (почему именно этот вариант наиболее актуален для вашего проекта).
- Основная часть:
  - Подробное описание процесса выполнения работы.
  - Содержимое файла конфигурации игнорирования (например, `.gitignore`).
  - Скриншоты, демонстрирующие результаты тестирования (например, вывод команды `git status`).
  - Описание возникших проблем и способов их решения.
- Заключение:
  - Выводы по работе (что нового узнали, какие навыки приобрели).
  - [Индивидуально:] Оценка эффективности настроек игнорирования для конкретной архитектуры проекта.
  - Возможные улучшения и дальнейшие шаги.