



# Технологии разработки программных приложений

Практическое занятие 1. 12 февраля

Михайлова Евгения Константиновна, ассистент кафедры МОСИТ  
mihajlova\_e@mirea.ru

# Структура курса

- Системы контроля и управления версиями ПО, документирование кода  
git, GitHub/GitLab
- Системы управления конфигурациями  
bash, переменные окружения, файлы конфигурации, регулярные выражения, системные процессы, плейбуки Ansible
- Системы сборки ПО  
системы управления пакетами, автоматизация сборки, Jenkins/GitLab CI
- Системы развертывания ПО  
аппаратная и контейнерная виртуализация, Docker, Kubernetes, непрерывная поставка ПО, подход «инфраструктура как код»

# Работа в семестре и баллы

За работу в семестре можно получить дополнительные баллы (до 25 баллов).

Предусмотрены следующие виды работ:

- 4 практические работы (оцениваются по 4 балла)
- Итоговый проект из 4 частей ( $2 + 2 + 2 + 2 = 8$  баллов)
- Коллоквиум в формате теста (1 балл)

У каждой практической работы есть мягкий и жесткий дедлайн. Итоговый проект разбит на 4 части (по 2 балла), дедлайны каждой части совпадают с дедлайнами соответствующих ПР.

После мягкого дедлайна урезается половина полученных за работу баллов

После жесткого дедлайна работу сдать невозможно (0 баллов)

# Таблица дедлайнов\*

Вид работы	Дедлайны	
	мягкий	жесткий
1 ПР	4 занятие	8 занятие
2 ПР	8 занятие	12 занятие
3 ПР	12 занятие	16 занятие
4 ПР	16 занятие	16 занятие
1-4 части Итогового проекта	Совпадают с дедлайнами практических работ 1-4	
Коллоквиум	Проводится на 8 занятии	

\* Сроки являются предварительными

# 1 практическая работа: «Системы контроля версий»

**Система управления версиями** (от англ. Version Control System, VCS или Revision Control System) — программное обеспечение для облегчения работы с изменяющейся информацией. Система управления версиями позволяет хранить несколько версий одного и того же документа, при необходимости возвращаться к более ранним версиям, определять, кто и когда сделал то или иное изменение, и многое другое.

## Задачи системы контроля версиями:

- фиксация изменений для отслеживаемых, добавляемых или удаляемых файлов исходного кода в рабочем дереве проекта;
- хранение истории зафиксированных изменений исходного кода;
- сравнение и поиск ранее зафиксированных изменений исходного кода;
- объединение наработок и экспериментальных изменений, ведущихся различными разработчиками над одним исходным кодом.

# 1 практическая работа: «Системы контроля версий»

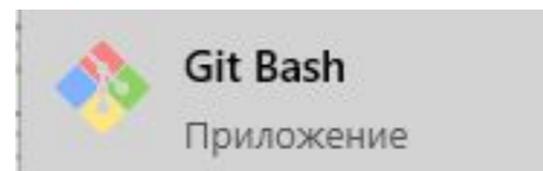
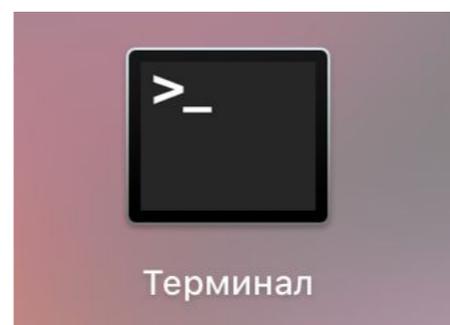
Задача на 1 пару:

пройти курс [githowto.com](https://githowto.com) до раздела 22.

Для студентов работающих на Windows необходимо использовать Git Bash для удобства работы с командами из курса.

Для студентов работающих на Linux/MacOS необходимо использовать командную строку системы.

Результат выполнения задач из курса необходимо включить в индивидуальный отчет (описания задач и скриншоты командной строки с выполненными операциями)



# 1 практическая работа: «Системы контроля версий»

В обязательном порядке должны быть выполнены следующие задачи:

1. Установите и настройте клиент git на своей рабочей станции.
2. Создайте локальный репозиторий и добавьте в него несколько файлов.
3. Внесите изменения в один из файлов.
4. Проиндексируйте изменения и проверьте состояние.
5. Сделайте коммит того, что было проиндексировано в репозиторий. Добавьте к коммиту комментарий.
6. Измените еще один файл. Добавьте это изменение в индекс git. Измените файл еще раз. Проверьте состояние и произведите коммит проиндексированного изменения. Теперь добавьте второе изменение в индекс, а затем проверьте состояние с помощью команды `git status`. Сделайте коммит второго изменения.
7. Просмотрите историю коммитов с помощью команды `git log`. Ознакомьтесь с параметрами команды и используйте некоторые из них для различного формата отображения истории коммитов.
8. Верните рабочий каталог к одному из предыдущих состояний.
9. Изучите, как создавать теги для коммитов для использования в будущем.
10. Отмените некоторые изменения в рабочем каталоге (до и после индексирования).
11. Отмените один из коммитов в локальном репозитории.

# Итоговый проект. 1 часть

Итоговый проект представляет из себя программное приложение, разработкой, конфигурированием, развертыванием и поставкой которого вы будете заниматься в семестре.

Тема проекта выбирается самостоятельно, но рекомендуется придерживаться следующих направлений:

- WEB-разработка
- Мобильная разработка
- Разработка под 1С
- Разработка базы данных
- Desktop приложение
- \*Интернет вещей (IoT)
- \*GameDev
- \*Разработка бот-сервиса

# Итоговый проект. 1 часть

## Состав команды:

- 2-3 человека в зависимости от сложности проекта
- Все члены команды будут выступать в роли разработчиков

Рекомендуется выбирать тему проекта, чтобы его непосредственная разработка заняла не более 15 часов на одного члена команды.

Приблизительную тему и идею проекта необходимо утвердить на текущей паре.