



Вводная лекция по ПИОГИ

ВСЕМ ПРИВЕТ С ВАМИ ДЕМОН И
АНДРОИД

3

3DS



MAX: общение,
полезные сервисы



– ЛОВИТ ДАЖЕ НА ЛЕКЦИИ

Что было \ что будет?

- Информатика (база)
- Пиоги I (ГУИ)
- Пиоги II (ООП+гит)

Контроль версий

Система контроля версий (СКВ, VCS - Version Control System) - это инструмент, который сохраняет историю изменений в файлах проекта (не обязательно чисто программирование), позволяя разработчикам отслеживать изменения и управлять версиями кода (или текста).

Основная цель СКВ - и упорядочить работу над проектом.

Что такое

Дано:

- Физика скоро
- 40 билетов
- Любой текстовой редактор

- Нанавижу гугл докс

Что такое

В Git каждый работает в своей локальной копии. Леха не может случайно затереть работу Светы. Он может затереть только свою. Все изменения сливаются вместе в общую версию осознанно, а не по факту нажатия клавиши.

Что такое

Три Состояния Ответа на Билет

1. Рабочий стол (Working Directory)
2. На проверку (Staging Area)
3. Архив (Repository)

Что такое

- `git init` – "Создать папку для подготовки": Леха создает папку `fizika_na_otlichno` и внутри нее выполняет эту команду. Все, теперь это не просто папка, а хранилище с историей.
- `git status` – "Что по билетам?": Леха вводит команду и видит: "`bilet_25.md` - изменен. `bilet_26.md` - добавлен в папку 'На проверку'. `bilet_27.md` - неотслеживаемый файл (ты его создал, но пока не смотрим)".

Что такое

- `git add bilet_25.md` – "Этот билет готов, кладу на проверку".
- `git commit -m "Сделал билет 25, устал как собака"` – "Сдать в архив".

Что такое

git log – "Посмотреть историю, кто что сдал": Леха вводит и видит:

commit <длинный_хэш_коммита_Светы>

Author: Святослав <sveta@email.com>

Date: Mon Sep 1 15:30:00 2025

Полностью расписала билеты 1-5. Уравнения Максвелла - огонь.

commit <длинный_хэш_коммита_Лехи>

Author: Lekha <lekha@email.com>

Date: Mon Sep 1 14:00:00 2025

Начал делать билеты 25-30. Добавил пустые файлы

Что такое

1. git clone <URL> – "Скачать все готовые билеты"
2. git pull – "Синхронизироваться":
3. git push – "Загрузить мои готовые билеты на GitHub":

Ветвление: Гениальная фича для спорных вопросов

Света говорит: "Я попробую свой вариант, но не буду трогать основной." Она создает "альтернативную реальность" (ветку): `git branch sveta-hardcore-mode`

Она переключается в нее: `git checkout sveta-hardcore-mode`

- Вариант А: Вариант Светы оказался шикарным.
- Вариант Б: Вариант Светы оказался слишком сложным.

А как правильно слияние делать?

Действие 1: Работа Светы

Билет 10: Законы Ньютона

Законы Ньютона – это три важнейших закона классической механики, которые позволяют записать уравнения движения для любой механической системы.

1. git add bilet_10_zakony_nyutona.md
2. git commit -m "Добавила определение к законам Ньютона в билет 10"
3. git push

А как правильно слияние делать?

Действие 2: Работа Лехи

Билет 10: Законы Ньютона

Короче, первый закон - это когда ты едешь в автобусе, он тормозит, а ты летишь вперед.

1. git add bilet_10_zakony_nyutona.md
2. git commit -m "Накидал пример для первого закона в 10 билете"
3. git push

А как правильно слияние делать?

Действие 3: Судный день

Git ему говорит: “My boy! На сервере уже есть изменения, которых у тебя нет. Сначала скачай их”.

Леха – послушный мальчик. Он пишет: `git pull`.

И тут...

А как правильно слияние делать?

БУМ! Git орет на него в консоли:

Auto-merging bilet_10_zakony_nyutona.md

CONFLICT (content): Merge conflict in bilet_10_zakony_nyutona.md

Automatic merge failed; fix conflicts and then commit the result.

А как правильно слияние делать?

Билет 10: Законы Ньютона

<<<<< HEAD

Короче, первый закон - это когда ты едешь в автобусе, он тормозит, а ты летишь вперед.

=====

Законы Ньютона – это три важнейших закона классической механики, которые позволяют записать уравнения движения для любой механической системы.

>>>>> a1b2c3d4e5f6... (хэш коммита Светы)

А как правильно слияние делать?

Билет 10: Законы Ньютона

Законы Ньютона – это три важнейших закона классической механики, которые позволяют записать уравнения движения для любой механической системы.

Например, первый закон - это когда ты едешь в автобусе, он резко тормозит, а ты по инерции продолжаешь лететь вперед.

А как правильно слияние делать?

1. git add bilet_10_zakony_nyutona.md
2. git commit (Можно без -m. Git сам откроет редактор и предложит сообщение типа "Merge branch 'main' of ...". Этого достаточно.)
3. git push.

Пулл-Реквест. Цивилизованный способ

Pull Requests (PR) или Merge Requests (MR)

Новое правило игры: Никто не пушит напрямую в main. Ветка main – священна.

Пулл-Реквест. Цивилизованный способ

1. Леха создает свою ветку: `git checkout -b lekha-nyuton-example`
2. Он делает свою работу в этой ветке, `git add`, `git commit`
3. `git push origin lekha-nyuton-example`
4. Create a Pull Request
5. Леха нажимает. Открывается форма, где он пишет:
 1. Заголовок: "Добавил пример к билету 10"
 2. Описание: "Эй, @Sveta, я тут накидал простой пример для первого закона. Глянь, плиз, норм или да? Может, как-то по-другому сформулировать?"

Пулл-Реквест. Цивилизованный способ

1. Все изменения, которые сделал Леха (подсвеченны зеленым).
2. Она может оставить комментарий к любой строчке кода: "Леха, пример про автобус – огонь, но давай добавим слово 'резко', так будет понятнее".
3. Главное: GitHub автоматически проверяет, можно ли слить эту ветку с main без конфликтов. Если Света за это время ничего не пушила в main, GitHub покажет зеленую галочку "Able to merge".
4. один из них нажимает большую зеленую кнопку "Merge Pull Request"
5. Гит сам вольет изменения, ветку lekha-pyuton-example можно удалить