

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Фурсов Владимир Алексеевич

Должность: И.О. директора Пятигорского института (филиал) Северо-Кавказского

федерального университета

Дата подписания: 08.06.2026 13:50:44

Уникальный программный ключ:

1c378726a41fd0143ae5bccc8ba81860b00daa62

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ

Колледж Пятигорского института (филиал) СКФУ

УТВЕРЖДАЮ

И.О. директора Пятигорского института
(филиал) СКФУ

В.А. Фурсов

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ЛАБОРАТОРНЫХ РАБОТ

ОП.04 ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ПРОФЕССИОНАЛЬНОЙ ДЕЯТЕЛЬНОСТИ

Специальности СПО

09.02.11 Разработка и управление программным обеспечением

Пятигорск 2026

Методические указания для лабораторных работ по дисциплине «Информационные технологии в профессиональной деятельности» составлены в соответствии с требованиями ФГОС СПО к подготовке выпуска для получения квалификации. Предназначены для студентов, обучающихся по специальности 09.02.11 Разработка и управление программным обеспечением

Лабораторная работа №1

Тема: ИТ электронной обработки данных.

Цель: Закрепление материала ИТ электронной обработки данных.

Оборудование: Программное обеспечение MS Windows.

Задание 1

1. В документе Microsoft Word создайте таблицу расходов коммерческой фирмы в соответствии с рисунком Рис. 12.1. Заполните столбцы и строки соответствующими формулами: первый столбец "Всего" - сумма значений за первый квартал, второй столбец "Всего" - сумма значений за второй квартал, столбец "Всего за полугодие" - сумма за первый и второй квартал, строка "Всего" - ежемесячная сумма всех статей расходов.

Таблица в документе Microsoft Word 2003.

Таблица расходов коммерческой фирмы

	A	B	C	D	E	F	G	H	I	J
1	Статьи расхода	I квартал			Всего	II квартал			Всего	Всего за полугодие
2		Янв.	Февр.	Март		Апр.	Май	Июнь		
3	Закупка	450,00	470,00	537,00		356,80	550,95	732,75		
4	Зарплата	417,70	547,05	555,00		348,00	605,00	800,80		
5	Реклама	260,00	211,00	237,00		704,80	205,00	301,60		
6	Аренда	82,00	73,00	77,50		70,00	69,50	85,77		
7	Командировки	75,00	94,00	100,00		82,78	106,90	123,29		
8	Коммун. услуги	17,00	19,00	20,80		15,33	17,99	20,44		
9	Экспл. расходы	12,00	13,00	11,50		10,22	14,00	15,55		
10	Всего									

Рис. 12.1

2. Установите параметры страницы документа Microsoft Word.

Установите следующие параметры страницы: размер бумаги - А4 (ширина - 21 см, высота - 29,7 см), поля (верхнее - 1,5 см; нижнее - 1,5 см; левое - 0,7 см; правое - 0,7 см).

3. Создайте таблицу расходов коммерческой фирмы в соответствии с Рис. 12.1.

Введите формулы вычислений в таблицу

Задание 2

1. Введите формулы вычислений суммы всех ячеек слева от заданной ячейки в столбец "Всего" за первый квартал.

Для ввода требуемой формулы установите курсор в ячейку E3 и выполните команду меню "Таблица/Формула...". В раскрывшемся окне диалога вставьте функцию =SUM() в строку "Формула" и введите аргумент функции LEFT [получите формулу с аргументом =SUM(LEFT)], выберите формат числа ###0,00.

2. Введите формулы вычислений суммы диапазона ячеек в столбец "Всего" за второй квартал

Для ввода этой формулы установите курсор в ячейку I3 и выполните команду меню "Таблица/Формула...". В раскрывшемся окне диалога вставьте функцию =SUM() в строку формула и введите аргумент функции F3:H3 [получите формулу с аргументом =SUM(F3:H3)], выберите формат числа # ##0,00.

3. Введите формулы вычислений суммы заданных ячеек в столбец "Всего за полугодие"

Для ввода необходимой формулы установите курсор в ячейку J3 и выполните команду меню "Таблица/Формула...". В раскрывшемся окне диалога вставьте функцию =SUM() в строку формула и введите аргумент функции E3:I3 [получите формулу с аргументом =SUM(E3:I3)], выберите формат числа # ##0,00.

4. Введите формулы вычислений суммы всех ячеек, расположенных выше заданной ячейки, в строку "Всего"

Для ввода данной формулы установите курсор в ячейку B10 (затем в остальные ячейки C10 - J10) и выполните команду меню "Таблица/Формула...". В раскрывшемся окне диалога вставьте функцию =SUM() в строку формула и введите аргумент функции ABOVE [получите формулу с аргументом =SUM(ABOVE)], выберите формат числа # ##0,00.

Контрольные вопросы:

1. Что такое данные?
2. Информационные технологии электронной обработки данных?

Лабораторная работа №2

Тема: ИТ управления. ИТ для создания и обработки деловых документов

Цель: Закрепление материала ИТ электронной обработки данных.

Информационные технологии экспертных систем

Характеристика и назначение. Наибольший прогресс среди компьютерных информационных систем отмечен в области разработки экспертных систем, основанных на использовании искусственного интеллекта. Экспертные системы

дают возможность менеджеру или специалисту получать консультации экспертов по любым проблемам, о которых этими системами накоплены знания.

Под *искусственным интеллектом* обычно понимают способности компьютерных систем к таким действиям, которые назывались бы интеллектуальными, если бы исходили от человека.

Решение специальных задач требует специальных знаний. Однако не каждая компания может себе позволить держать в своем штате экспертов по всем связанным с ее работой проблемам или даже приглашать их каждый раз, когда

проблема возникла. Главная идея использования технологии экспертных систем

заключается в том, чтобы получить от эксперта его знания и, загрузив их в память компьютера, использовать всякий раз, когда в этом возникнет необходимость.

Являясь одним из основных приложений искусственного интеллекта, экспертные системы представляют собой компьютерные программы, трансформирующие опыт

экспертов в какой-либо области знаний в форму эвристических правил (эвристик).

Эвристики не гарантируют получения оптимального результата с такой же уверенностью, как обычные алгоритмы, используемые для решения задач в рамках технологии поддержки принятия решений. Однако часто они дают в достаточной

степени приемлемые решения для их практического использования. Все это делает возможным использовать технологию экспертных систем в качестве советующих систем.

Сходство информационных технологий, используемых в экспертных системах и системах поддержки принятия решений, состоит в том, что обе они обеспечивают

высокий уровень поддержки принятия решений. Однако имеются три существенных

различия. *Первое* связано с тем, что решение проблемы в рамках систем поддержки принятия решений отражает уровень ее понимания пользователем и его

возможности получить и осмыслить решение. Технология экспертных систем, наоборот, предлагает пользователю принять решение, превосходящее его возможности. *Второе* отличие указанных технологий выражается в способности экспертных систем пояснять свои рассуждения в процессе получения решения. Очень часто эти пояснения оказываются более важными для пользователя, чем само решение. *Третье* отличие связано с использованием нового компонента информационной технологии – знаний.

Основными компонентами информационной технологии, используемой в экспертной системе, являются (рис. 1): интерфейс пользователя, база знаний, интерпретатор, модуль создания системы.

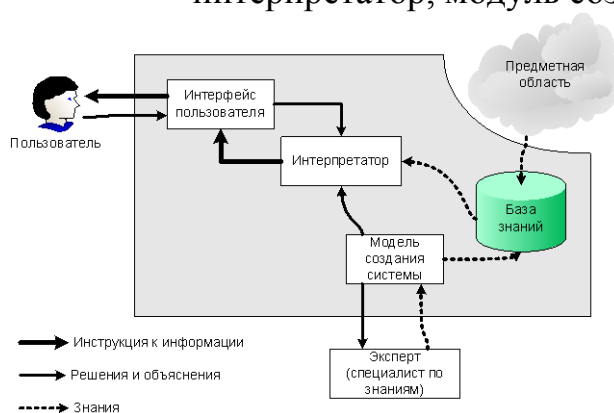


Рис. 1. Основные компоненты информационной технологии экспертных систем

Система поддержки принятия решений – диалоговая автоматизированная информационная система, использующая правила решений и соответствующие модели с базами данных, а также интерактивный компьютерный процесс моделирования, поддерживающий принятие самостоятельных и

неструктурированных решений отдельными менеджерами и личным опытом лица, принимающего решения, для получения конкретных, реализуемых решений проблем, не поддающихся решению обычными методами.

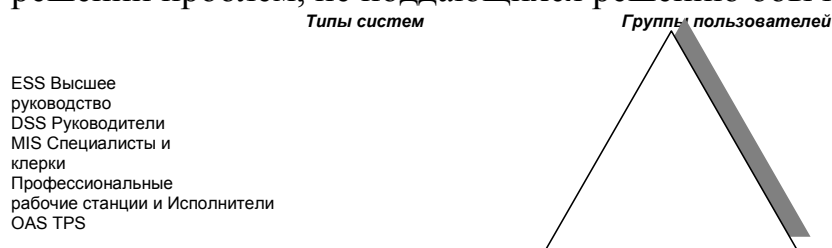


Рис. 2. Пользователи различных типов информационных систем

В последнее время СППР начинают применяться и в интересах малого и среднего бизнеса (например выбор варианта размещения торговых точек, выбор кандидатуры на замещение вакантной должности, выбор варианта информатизации и т.д.). В общем, они способны поддержать индивидуальный стиль и соответствовать персональным потребностям менеджера.

Существуют системы, созданные для решения сложных проблем в больших коммерческих и государственных организациях:

Система авиалиний. В отрасли авиаперевозок используется система поддержки принятия решений – Аналитическая Информационная Система Управления. Она была создана *American Airlines*, но используется и остальными компаниями, производителями самолетов, аналитиками авиаперевозок, консультантами и ассоциациями. Эта система поддерживает множество решений в этой отрасли путем анализа данных, собранных во время утилизации транспорта, оценки грузопотока, статистического анализа графика. Например, она позволяет делать прогнозы для авиарынка по долям компаний, выручке и рентабельности. Таким образом, эта система позволяет руководству авиакомпании принимать решения относительно цены билетов, запросов в транспорте и т.д.

Географическая система. Географическая информационная система – это специальная категория систем поддержки, которая позволяет интегрировать компьютерную графику с географическими БД и с другими функциями систем

поддержки принятия решений. Например, *IBMs GeoManager* – это система, которая позволяет конструировать и показывать карты и другие визуальные объекты для

помощи при принятии решений относительно географического распределения людей и ресурсов. Например, она позволяет создать географическую карту преступности и помогает верно перераспределить силы полиции. Также ее используют для изучения степени урбанизации, в лесной промышленности, железнодорожном бизнесе и т.д.

Контрольные вопросы

1. Перечислите виды информационных технологий по степени охвата задач управления.
2. Характеристика и назначение ИТ обработки данных.
3. Основные компоненты ИТ обработки данных.
4. Характеристика и назначение ИТ управления.
5. Основные компоненты ИТ управления.
6. Какие задачи относятся к офисным?
7. Что называют электронным офисом?
8. Характеристика и назначение ИТ автоматизации офиса.
9. Основные компоненты ИТ автоматизации офиса.
10. Характеристика и назначение ИТ поддержки принятия решений.
11. Основные компоненты ИТ поддержки принятия решений.
12. Что является главной особенностью информационной технологии поддержки принятия решений?

13. Какими возможностями должна обладать система управления базой моделей (СУБМ)?
14. Из каких моделей состоит база моделей в системах поддержки принятия решения?
15. Характеристика и назначение ИТ экспертных систем.
16. Основные компоненты ИТ экспертных систем.
17. Что обычно понимают под искусственным интеллектом?
18. В чем состоит сходство информационных технологий, используемых в экспертных системах и системах поддержки принятия решений?
19. Что содержится в базе знаний?
20. Какова эволюция систем поддержки принятия решений?

Лабораторная работа №3

Тема: Сравнение работы нескольких ИИ инструментов

Цель работы: изучить пользовательский интерфейс и основы работы в Google Colab.

Перечень обеспечивающих средств

Задания лабораторной работы выполняются в любом браузере.

Общие теоретические сведения

Jupyter Notebook (Jupyter-ноутбук) – это среда разработки, где сразу можно видеть результат выполнения кода и его отдельных фрагментов. Отличие от традиционной среды разработки в том, что код можно разбить на куски и выполнять их в произвольном порядке.

В такой среде разработки можно, например, написать функцию и сразу проверить ее работу, без запуска программы целиком. А еще можно поменять порядок выполнения кода. Можно отдельно загрузить файл в память, отдельно проверить его содержимое, отдельно обработать содержимое.

Файлы, созданные в Jupyter Notebook, имеют расширение `ipynb` и называются блокнотами.

Также в jupyter-ноутбуках (далее будем называть их просто ноутбуками или блокнотами) есть вывод результата сразу после фрагмента кода. Например, можно прямо в середине кода увидеть построенный график, получить предварительные цифры или любую другую визуализацию.

Чаще всего, когда говорят о ноутбуках, имеют в виду работу с Python. Но на самом деле можно работать и с другими языками: Ruby, Perl, R, bash-скрипты.

Основная область применения ноутбуков – машинное обучение, нейросети, визуализация данных и статистика.

Еще такую среду часто используют для поэтапной разработки, когда нужно по шагам проверять работу разных фрагментов кода. Дело в том, что код в ноутбуках хранится в независимых ячейках и его можно запускать в любом порядке или поодиночке. Это позволяет быстро экспериментировать с алгоритмами и находить оптимальное решение.

Также ноутбуки используются при обучении программированию на Python, чтобы писать код и сразу видеть результат его работы.

С файлами Jupyter Notebook можно работать и с помощью онлайн-сервиса Google Colaboratory.

Google Colaboratory (Google Colab) – это облачный сервис, направленный на упрощение исследований в области машинного и глубокого обучения.

Главная особенность Google Colab – бесплатные мощные графические процессоры GPU (Graphics Processing Unit, графический процессор) и TPU (Tensor Processing Unit, тензорный процессор), благодаря которым можно заниматься не только базовой аналитикой данных, но и более сложными исследованиями в области машинного обучения. С тем, что CPU (Central Processing Unit, центральный процессор) вычисляет часами, GPU или TPU справляются за минуты или даже секунды.

CPU – центральный процессор – мозг компьютера, который выполняет операции с данными. Настолько универсален, что может использоваться почти для всех задач: от записи фотографий на флешку до моделирования физических процессов.

GPU – графический процессор. Обработывает данные быстрее, так как задачи выполняет параллельно, а не последовательно, как CPU. Он заточен исключительно под графику, поэтому на нем удобнее работать с изображением и видео, например, заниматься 3D-моделированием или монтажом.

TPU – тензорный процессор, разработка Google. Он предназначен для тренировки нейросетей. У этого процессора в разы выше производительность при больших объемах вычислительных задач.

Сами процессоры дорогие, и не каждый может их себе позволить. Google Colaboratory дает возможность бесплатно и непрерывно пользоваться ими на протяжении 12 часов. Будьте внимательны: как только это время истечет, Colab сотрет все данные и придется начинать сначала.

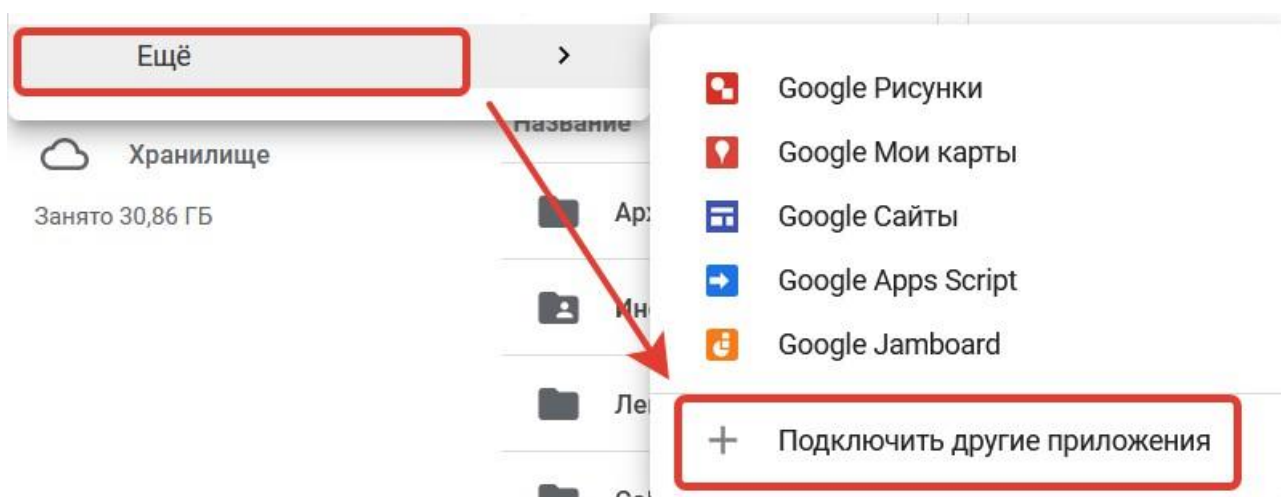
Кроме того, Google отключает блокноты после примерно 30 минут бездействия, чтобы не перегружать процессоры. Система Colab так устроена специально: многие факторы, в том числе время простоя, максимальная активность, общие ограничения на объем памяти иногда динамически меняются. Активным участникам ненадолго могут ограничить доступ к GPU, чтобы дать возможность использовать процессор другим.

Google Colab максимально упростил все процессы: в нем есть и базовые библиотеки (NumPy, scikit-learn, Pandas), и более сложные (вроде Keras, TensorFlow или PyTorch), не нужно ставить программы и среды самостоятельно, можно просто сразу писать код. Если же базовых библиотек недостаточно, всегда можно добавить необходимые с помощью установщика PIP.

Для работы с Colab нужно иметь аккаунт Google. Если у вас его нет, создайте его.

Поскольку все блокноты хранятся в облачном хранилище Google, начнем работу со страницы [Google Диск](#). Чтобы упорядочить блокноты рекомендуется хранить их в одной папке, например, в Colab Notebooks.

Далее нужно установить приложение Google Colaboratory в среду Google Диск (далее просто Диск). Для этого нажмем на кнопку Создать в левом верхнем углу на странице Диска. Выберем пункт Еще – Подключить другие приложения.



Затем в списке выберем Google Colaboratory.



Нажмем кнопку Установить. Дадим разрешение приложению на установку, нажав Продолжить. Выберем аккаунт Google, в который будет установлено приложение.

После установки согласимся, что Google Colaboratory будет приложением по умолчанию для файлов Jupyter Notebook.

В качестве альтернативы можно открыть Colab по ссылке <https://colab.research.google.com>.

Создадим свой первый блокнот через Диск. Для этого нажмем кнопку Создать, затем в списке выберем Еще, затем Google Colaboratory. Откроется основное окно интерфейса Colab. Рассмотрим его подробно.

В левом верхнем углу находится логотип Colab. Он является ссылкой. Нажав на него, откроется Диск, в котором будут показан список ноутбуков, созданных в Colab. В этом списке можно быстро открыть нужный блокнот.

Справа от логотипа указано название блокнота, которое можно сразу изменить. Файл был сохранен автоматически в текущей папке в момент запуска Colab.

Справа находится кнопка Подключиться, нажав на которую мы подключимся к серверу Google, на котором будет выполняться наш код. После подключения будет указана статистика по ОЗУ и месту на диске.



В блокноте вся информация размещается в ячейках. Есть два типа ячеек: кодовые ячейки и текстовые ячейки. Текстовые ячейки обычно используются для добавления комментариев к коду, пояснениям, а также для отображения изображений, видео, заголовков, списков и многого другого. Для форматирования данных в текстовых ячейках используется Markdown – облегченный язык разметки, созданный с целью обозначения форматирования в простом тексте, с максимальным сохранением его читаемости человеком, и пригодный для машинного преобразования в языки для продвинутых публикаций.

Чтобы добавить кодовую или текстовую ячейку в Colab нужно выбрать соответствующий пункт в меню Вставка. Альтернативный способ – подвести указатель мыши к середине нижней границы ячейки и выбрать одну из кнопок: + Код или + Текст.

Чтобы запустить текущую ячейку кода нужно нажать клавиши Ctrl+Enter. Нажатие сочетания Shift+Enter приведет к запуску ячейки и переходу к ячейке ниже.

Задания

Задание 1. Изучить краткое официальное руководство по работе с Google Colab. Для этого перейти по ссылке <https://colab.research.google.com/?hl=ru> и изучить представленный материал. Сохраните копию этого блокнота на своем Диске.

Задание 2. Создать новый блокнот. В первой кодовой ячейке напишите код:
`print("Hello, World!")`

Запустите текущую ячейку, посмотрите результат.

Задание 3. В следующей ячейке первой переменной присвоить числовое значение, равное порядковому номеру первой буквы вашей фамилии. В качестве имени переменной задать свою фамилию, написанную латиницей. В следующей строке второй переменной присвоить числовое значение, равное порядковому номеру второй буквы вашей фамилии. В качестве имени переменной задать название вашей группы, написанное латиницей.

В следующей строке сложить обе переменные, используя оператор «+».

Запустите текущую ячейку, посмотрите результат.

Ниже представлен пример выполнения.



The screenshot shows a code cell in Google Colab. On the left is a play button icon. The code inside the cell is:
`Katsko = 12`
`KafPI = 1`
`Katsko + KafPI`
Below the code, the result of the last line is displayed as `13`.

Справа от ячейки появилась панель с инструментами. Изучите ее.

Задание 4. Создать новую текстовую ячейку.

Ввести следующий текст в текстовую ячейку:

This is ****bold****.

This is **italic**.

This is ~~~strikethrough~~~.

В правой части ячейки вы увидите результат форматирования текста.

This is **bold**. This is *italic*. This is ~~strikethrough~~.

Часто используемые настройки форматирования расположены в верхней части ячейки.

Создайте новую текстовую ячейку, добавьте в нее произвольный текст и отформатируйте его, используя полужирное начертание, выделение курсивом и зачеркнутый текст. Запустите текущую ячейку, посмотрите результат.

Задание 5. Ввести следующий текст в новую текстовую ячейку:

$\sqrt{3x-1}+(1+x)^2$

В правой части ячейки вы увидите результат форматирования текста.

$$\sqrt{3x-1}+(1+x)^2$$

Создайте новую текстовую ячейку. Используйте возможности разметки markdown, чтобы создать собственную формулу, включающую квадратный корень, степень числа и несколько арифметических операций. Запустите текущую ячейку, посмотрите результат.

Задание 6. Настроить совместный доступ к созданному блокноту.

Для этого нужно нажать на кнопку Поделиться, расположенную сверху справа.

Установить уровень доступа: Просматривать могут все в Интернете, у кого есть эта ссылка.



Скопируйте ссылку

Просматривать могут все в Интернете, у кого есть эта ссылка.

[Изменить](#)

[Копировать ссылку](#)

Проверьте, что установлен уровень совместного доступа Читатель.



Доступные пользователям, у которых есть ссылка ▾

Просматривать могут все в Интернете, у кого есть эта ссылка.

Читатель ▾

Скопируйте ссылку для получения общего доступа и поместите ее в отчет.

[https://colab.research.google.com/drive/1oqeLPnNhFi-3uQViwb ...](https://colab.research.google.com/drive/1oqeLPnNhFi-3uQViwb...)

[Копировать ссылку](#)

Нажмите Готово.

Содержание отчета

1. Титульный лист
2. Цель работы.
3. Формулировка задания и краткое описание результатов выполнения каждого задания.
4. Ссылка общего доступа к созданному блокноту.
5. Общий вывод о проделанной работе.

Лабораторная работа №4

Тема: Подключение и использование ChatGPT для генерации кода.

Создание readme-файла проекта через ИИ

За последние пару лет ИИ для программирования перестал быть “игрушкой для автодополнения” и превратился в рабочий инструмент: он помогает проектировать архитектуру, писать тесты, ускорять ревью и разбирать чужой код без боли. Но есть нюанс: чтобы ИИ реально давал пользу, а не “красивый, но бесполезный текст”, нужен понятный процесс, правила качества и грамотные запросы.

Разберёмся, **как выжать максимум из Chat GPT** в реальной разработке: от постановки задач и промптов до безопасности и контроля качества. Будем говорить на языке коммерческих проектов: сроки, стабильность, читаемость и поддерживаемость.

Модель не заменяет инженера — она усиливает инженера. И в этом ключе появляется новый образ: **ИИ-программист** как “вторые руки”, которые выполняют рутинные куски, предлагают варианты, находят ошибки и ускоряют итерации.

Если вы хотите попробовать режим “помощника-разработчика” в структурированном формате, посмотрите **Chat GPT для разработчиков** — это удобная точка входа, когда нужен не просто чат, а ассистент под задачу.

Почему ИИ в кодинге — это не магия, а технология процесса

Главный миф: “ИИ пишет код, значит программист не нужен”. В реальности всё наоборот: чем сложнее система, тем важнее инженерный контроль. ИИ генерирует гипотезы и заготовки; разработчик превращает их в продукт.

Вот что реально меняется, когда вы подключаете AI коддинг в командную практику:

- меньше времени на шаблонный код и бойлерплейт;
- быстрее анализ требований и декомпозиция;
- проще поддерживать единый стиль и соглашения;
- быстрее разбирать баги и регрессии;
- ускоряется обучение джунов и онбординг.

Эффект часто выражается не только в “сэкономленных часах”, но и в снижении когнитивной нагрузки: меньше переключений контекста, меньше тупиков, быстрее решения.

Где ИИ приносит максимум пользы в коммерческой разработке

1) Генерация заготовок и типового кода

Генерация кода GPT лучше всего работает там, где задача хорошо формализуется: CRUD, схемы DTO, слой репозитория, сериализация, адаптеры API, обвязка логирования, конфиги.

Пример: вы задаёте контракт входных/выходных данных, ограничения и стиль проекта — и получаете стартовую реализацию, которую доводите до продакшен-качества.

2) Ускорение работы с легаси

В легаси основное время уходит на “понять”. ИИ помогает:

- объяснить участок кода простыми словами;

- подсветить потенциальные дефекты и точки риска;
- предложить стратегию рефакторинга;
- найти места, где тесты критичны.

Это типичная помощь программисту, которая экономит часы на чтении и разборе.

3) Тестирование и качество

ИИ полезен для:

- генерации набора тест-кейсов;
- написания юнит-тестов по спецификации;
- формирования edge cases и негативных сценариев;
- подготовки моков и фикстур.

Если вы правильно задаёте “границы поведения”, вы получаете не просто тесты, а покрытие логики.

4) Ревью и поддерживаемость

ИИ умеет выступать как “предварительный ревьюер”:

- найти нарушения соглашений;
- предложить упрощение;
- подсказать опасные конструкции;
- сформулировать комментарии к PR.

Результат: ускорение разработки без просадки качества, если ревью остаётся у человека.

Какой результат считать “хорошим”, когда код пишет ИИ

Чтобы умный код не превратился в “умные ошибки”, задайте критерии приёмки. Минимальный чек-лист:

- код компилируется/собирается и проходит тесты;
- нет новых предупреждений линтера;
- соблюдены соглашения проекта (нейминг, структура, DI, ошибки);
- учитываются нефункциональные требования (производительность, безопасность);
- добавлены тесты на ключевые ветки;
- код читаем и поддерживаем.

ИИ может быстро накидать решение, но именно вы определяете, проходит ли оно через “ворота качества”.

Роли Chat GPT в разработке: не один инструмент, а несколько режимов

Полезно мыслить не “чатом”, а ролями:

Архитектор

Помогает выбрать подходы, паттерны, границы модулей, контракты, интеграции.

Реализатор

Пишет код по техническому заданию, создаёт файлы, скелеты, повторяющиеся компоненты.

Тест-инженер

Генерирует тест-кейсы, тесты, проверяет ветвления, думает о негативных сценариях.

Ревьюер

Смотрит на читаемость, безопасность, крайние случаи, стиль.

Документатор

Пишет README, комментарии, ADR, инструкции по запуску и деплою.

Так вы получаете не “магический генератор”, а управляемую автоматизацию разработки через разделение задач.

Практика: как правильно ставить задачу ИИ, чтобы получить рабочий результат

Если вы хотите, чтобы ИИ для программирования выдавал качественный код, ему нужен контекст. Хороший запрос обычно содержит 6 блоков:

1. Цель: что делаем и зачем.
2. Входные данные: форматы, примеры, ограничения.
3. Выходные данные: структура ответа, формат, сигнатуры.
4. Окружение: язык, фреймворк, версии, архитектура проекта.
5. Ограничения: стиль, производительность, безопасность, запреты.
6. Критерии готовности: тесты, линтер, edge cases.

Мини-шаблон промпта (под большинство задач)

- Язык/стек: ...
- Контекст проекта: ...
- Задача: ...
- Требования: ...
- Ограничения: ...
- Выдай: (список файлов/функций/инструкций)
- Добавь тесты: да/нет, какие именно
- Объясни решения кратко: да/нет

С таким шаблоном AI кодирование становится предсказуемым.

Типовые сценарии использования: от идеи до PR

Сценарий 1. Быстрый старт фичи

1. Декомпозиция задачи вместе с ИИ.
2. Проектирование API/контрактов.
3. Генерация каркаса.
4. Доработка под стандарты проекта.
5. Написание тестов.
6. Саморевью “второй парой глаз” через ИИ.

Здесь ускорение разработки часто самое заметное: меньше пауз и тупиков.

Сценарий 2. Багфикс с минимальным риском

1. Дайте ИИ симптомы, логи, контекст и ожидаемое поведение.
2. Попросите список гипотез (5–10 причин).
3. Попросите план проверки гипотез.
4. Попросите патч и тест, который ловит баг.
5. Проверьте руками: воспроизведение → фикс → регресс.

Это особенно эффективно, когда баг “не очевидный”, а время горит.

Сценарий 3. Рефакторинг и снижение техдолга

1. Попросите оценить проблемные зоны: дублирование, цикломатика, связность.
2. Сформируйте план рефакторинга маленькими шагами.
3. Начните с тестов или snapshot поведения.
4. Делайте мелкие PR.

ИИ помогает видеть варианты, но ответственность за дизайн — на вас.

Что именно можно делегировать ИИ, а что лучше оставить человеку

Отлично делегируется

- шаблонные куски и повторяемые паттерны;
- преобразование форматов (DTO ↔ domain ↔ persistence);
- генерация тестов по описанию поведения;
- документирование и примеры использования;
- миграции и “переклейка” кода при смене API;
- варианты рефакторинга.

Лучше контролировать вручную

- архитектурные решения с долгосрочными последствиями;
- безопасность, авторизация, криптография, обработка секретов;
- оптимизация горячих путей и сложная производительность;
- финальные решения по бизнес-логике.

Важно помнить: ИИ-программист хорош как исполнитель и советник, но не как “единственный автор”.

Как получать качественный код: техники промптинга для разработчика

1) Просите варианты, а не единственный ответ

Фраза “дай 3 решения с плюсами/минусами” резко повышает качество.

2) Просите уточняющие вопросы

Скажите: “Если данных не хватает — задай до 5 вопросов”. Это снижает риск не того решения.

3) Просите сначала план, потом код

План помогает вам быстро поймать ошибки дизайна до написания файлов.

4) Задавайте стиль и правила проекта

Например: “без глобального состояния”, “ошибки через Result”, “логирование через интерфейс”, “только async/await”.

5) Просите тесты и проверки

“Добавь юнит-тесты и перечисли edge cases” — обязательная часть для коммерческого качества.

Ошибки, которые чаще всего делают команды при внедрении ИИ

Ошибка 1. Отсутствие процесса контроля

Если нет линтера, тестов и ревью, генерация кода GPT просто ускорит поставку дефектов.

Ошибка 2. Слишком большой кусок за раз

ИИ лучше работает небольшими итерациями: файл/модуль/функция, а не “перепиши весь сервис”.

Ошибка 3. Слепая вера в “правильность”

ИИ может уверенно ошибаться. Поэтому проверяем:

- компиляция/сборка;
- тесты;
- статический анализ;
- безопасность;
- пограничные случаи.

Ошибка 4. Утечки контекста и данных

Нельзя вставлять в запросы приватные ключи, токены, пароли, персональные данные клиентов и внутренние секреты. Введите правила, что разрешено, а что нет.

Безопасность: как использовать ИИ и не подставить проект

Коммерческая разработка всегда про риск. Если вы используете ИИ для программирования в компании, зафиксируйте политику:

- Не передавать секреты (ключи, токены, приватные URL с доступом).
- Не отправлять персональные данные.
- Редактировать логи: маскировать идентификаторы, email, номера.
- Описывать проблему на синтетических примерах.
- Использовать обезличенные фрагменты кода, если есть ограничения.

Также полезно иметь “красную линию”: какие классы задач запрещены для ИИ (например, криптография и безопасность без участия ведущего инженера).

Как измерить эффект: метрики внедрения ИИ в разработку

Чтобы автоматизация разработки была управляемой, измеряйте не “ощущения”, а показатели:

- время цикла задачи (от “в работе” до “в проде”);
- скорость закрытия багов (MTTR);
- количество багов на релиз;
- качество ревью (количество замечаний и повторных правок);
- доля задач, где ИИ помог (самоотчёты);
- время онбординга новых разработчиков.

Иногда прирост скорости проявляется не сразу: сначала команда учится писать запросы и выстраивает контроль качества.

Как встроить ИИ в командный процесс: практические правила

Ниже — набор правил, которые обычно “выстреливают” в коммерческих командах.

Правило 1. ИИ пишет — CI проверяет

Любой код, полученный через AI коддинг, обязан проходить те же ворота: тесты, линтер, сборка.

Правило 2. Маленькие PR

Делайте PR по 200–400 строк (если возможно). Так проще ревью, меньше рисков.

Правило 3. ИИ как ассистент, не как автор

Хороший подход: “ИИ предложил — разработчик утвердил и несёт ответственность”.

Правило 4. Единый набор промптов для команды

Создайте “промпт-гайд”: как задавать контекст, как описывать стиль, как просить тесты.

Правило 5. Обязательные тесты для логики

Если ИИ написал бизнес-логику — тесты обязательны. Это самая надёжная страховка.

Примеры задач, где ИИ особенно полезен

Быстрая генерация вспомогательных утилит

- парсеры и валидаторы;

- конвертеры форматов;
- мапперы сущностей;
- генераторы мок-данных.

Документация и примеры использования API

- примеры запросов;
- описание эндпоинтов;
- инструкции для фронтенда;
- контрактные примеры.

Подготовка миграций и аккуратных изменений

ИИ помогает составить план миграции схемы или перехода на новый интерфейс без “сломать всё”.

Если вы хотите, чтобы ассистент “держал рамки” и выдавал результат в формате разработчика (с проверками и пошаговыми действиями), попробуйте генерация кода GPT в режиме, где запросы и ответы ориентированы на практику.

“Умный код” — это не про сложность, а про читаемость и поддержку

Частая ловушка: ИИ предлагает более “хитрый” код, чем нужен. В коммерческой разработке ценится не изящество, а предсказуемость.

Признаки хорошего результата:

- код легко читать по верхнему уровню;
- функции короткие, нейминг ясный;
- ошибки обработаны явно;
- нет скрытых побочных эффектов;
- логика покрыта тестами;
- входы/выходы задокументированы.

Если ИИ предлагает слишком сложную конструкцию — попросите переписать “проще, но не медленнее”.

Как использовать ИИ в разных ролях разработчика

Для бэкенд-разработчика

- генерация контрактов API и DTO;
- обработка ошибок и статусы;
- структурирование доменной логики;
- тесты (юнит/интеграционные);
- оптимизация запросов (в рамках разумного, с проверками).

Для фронтенд-разработчика

- каркас компонентов;
- типизация и интерфейсы;
- обработка состояний и крайние случаи UI;
- тесты компонентов;
- генерация моков и стора.

Для мобильной разработки

- шаблоны экрана и навигации;
- работа с сериализацией;
- обработка ошибок сети;
- тесты ViewModel/Presenter.

Для DevOps/Platform

- шаблоны CI/CD;
- проверки качества;
- генерация конфигов;
- документация пайплайнов.

Помощь программисту тут часто не в “написать всё”, а в “не забыть важное” и “быстро собрать правильный каркас”.

Реальные ограничения и как их обходить

Ограничение 1. Недостаток контекста

Решение: давать минимально достаточный контекст — интерфейсы, примеры входов/выходов, соглашения проекта.

Ограничение 2. Возможные галлюцинации

Решение: просить ссылки на конкретные места в коде, тесты, проверяемые утверждения и пошаговую проверку.

Ограничение 3. “Слишком уверенный” ответ

Решение: просить альтернативы и явно требовать перечислить допущения.

Ограничение 4. Несовпадение со стилем команды

Решение: закрепить “код-стандарт” в промпте, просить следовать линтеру и паттернам проекта.

Продвинутые техники: как выжать максимум из Chat GPT

Техника “Контракт → реализация → тест”

1. Сначала попросите определить контракт.
2. Потом — реализацию по контракту.
3. Потом — тесты по контракту.

Так вы снижаете риск расплывчатых решений и получаете проверяемый результат.

Техника “Сначала критика”

Попросите ИИ сначала раскритиковать своё решение: “Найди 10 проблем в предложенном подходе”. Часто это выявляет слабые места.

Техника “Роль + ограничения”

Например: “Ты строгий ревьюер, ищешь проблемы безопасности и гонки данных”. Ролевой режим помогает фокусировать ответ.

Техника “Маленькие шаги”

Вместо “сделай всё” — “сделай часть А, затем часть В”. Для коммерческих задач это самый надёжный способ.

Какой стек лучше “дружит” с ИИ

ИИ обычно лучше показывает себя там, где:

- есть чёткие соглашения и структура проекта;
- логика отделена от инфраструктуры;
- покрытие тестами хотя бы базовое;
- CI настроен и ловит ошибки рано.

Если хаос в архитектуре, ускорение разработки будет нестабильным: ИИ начнёт “угадывать”, а не решать.

Влияние на команду: джуны, мидлы, сеньоры

Джуны

ИИ ускоряет обучение: объяснения, примеры, чек-листы. Но важно не превращать процесс в “копипаст без понимания”. Задача тимлида — научить проверять и думать.

Мидлы

Получают максимальную отдачу: они уже понимают архитектуру и умеют оценить качество, но всё ещё тратят много времени на рутину. Здесь AI коддинг даёт сильный прирост.

Сеньоры

Чаще используют ИИ для архитектурных обсуждений, ревью, поиска рисков и улучшения документации. ИИ работает как ускоритель принятия решений, а не как писатель кода “вместо”.

Чек-лист внедрения ИИ в проект за 7 шагов

1. Определите задачи, где ИИ даст пользу (рутина, тесты, документация).
2. Зафиксируйте правила безопасности (что нельзя отправлять).
3. Создайте базовый шаблон промпта и код-стандарты.
4. Подключите CI: линтер + тесты.
5. Начните с маленьких задач и измерьте эффект.
6. Сформируйте библиотеку “хороших промптов” для команды.
7. Регулярно пересматривайте практики по результатам метрик.

Это превращает автоматизация разработки из хаотичного эксперимента в системный процесс.

Частые вопросы и сомнения

“Не станет ли код хуже?”

Станет, если нет контроля качества. При наличии тестов, линтера и ревью код обычно становится стабильнее: меньше забытых проверок, больше единообразия, быстрее правки.

“Как не потерять инженерное мышление?”

Используйте ИИ как помощника, а не как “замену”. Просите объяснить решения, спорьте, просите альтернативы и обоснуйте выбор. Тогда ИИ-программист будет тренажёром мышления, а не костылём.

“Можно ли доверять ИИ в продакшене?”

Доверять можно только проверенному: тестами, ревью, мониторингом. ИИ не отменяет инженерную ответственность.

FAQ

1) Чем отличается ИИ для программирования от обычного автокомплита?

Автокомплит дополняет строку, а ИИ для программирования помогает на уровне задач: предлагает архитектуру, генерирует компоненты, пишет тесты, объясняет легаси и подсказывает стратегии.

2) Когда генерация кода GPT наиболее эффективна?

Когда задача хорошо описана и есть ограничения: контракты, примеры данных, стиль проекта. В таких условиях генерация кода GPT даёт быстрый каркас и экономит время на рутине.

3) Как безопасно использовать Chat GPT для разработчиков в коммерческом проекте?

Не отправляйте секреты и персональные данные, маскируйте логи, используйте обезличенные примеры, фиксируйте правила команды. Chat GPT для разработчиков должен работать внутри понятных рамок.

4) Можно ли поручать ИИ написание бизнес-логики?

Можно, но только с обязательными тестами и ревью. ИИ хорош как черновик и генератор вариантов, но финальная проверка — за человеком. Так помощь программисту приносит пользу без риска.

5) Как быстрее всего почувствовать ускорение разработки?

Начните с типовых задач: тесты, DTO/маппинг, документация, рефакторинг небольших модулей. При правильной постановке задач ускорение разработки заметно уже в первые недели.

Итоги: как превратить ИИ в конкурентное преимущество команды

ИИ в кодировании — это не “кнопка написать продукт”, а рычаг, который усиливает вашу инженерную систему. Если у вас есть стандарты, проверки и понятная постановка задач, умный код появляется быстрее, а качество становится стабильнее.

Запомните простой принцип: ИИ ускоряет то, что уже выстроено. Поэтому сначала — процесс и контроль, потом — масштабирование. Тогда AI кодирование станет не экспериментом, а устойчивой практикой, которая приносит бизнес-результат.

Лабораторная работа №5

Тема: Регистрация и запуск виртуальной машины в Яндекс.Облаке

Цель работы: Создание виртуальной машины при помощи Yandex Compute Cloud.

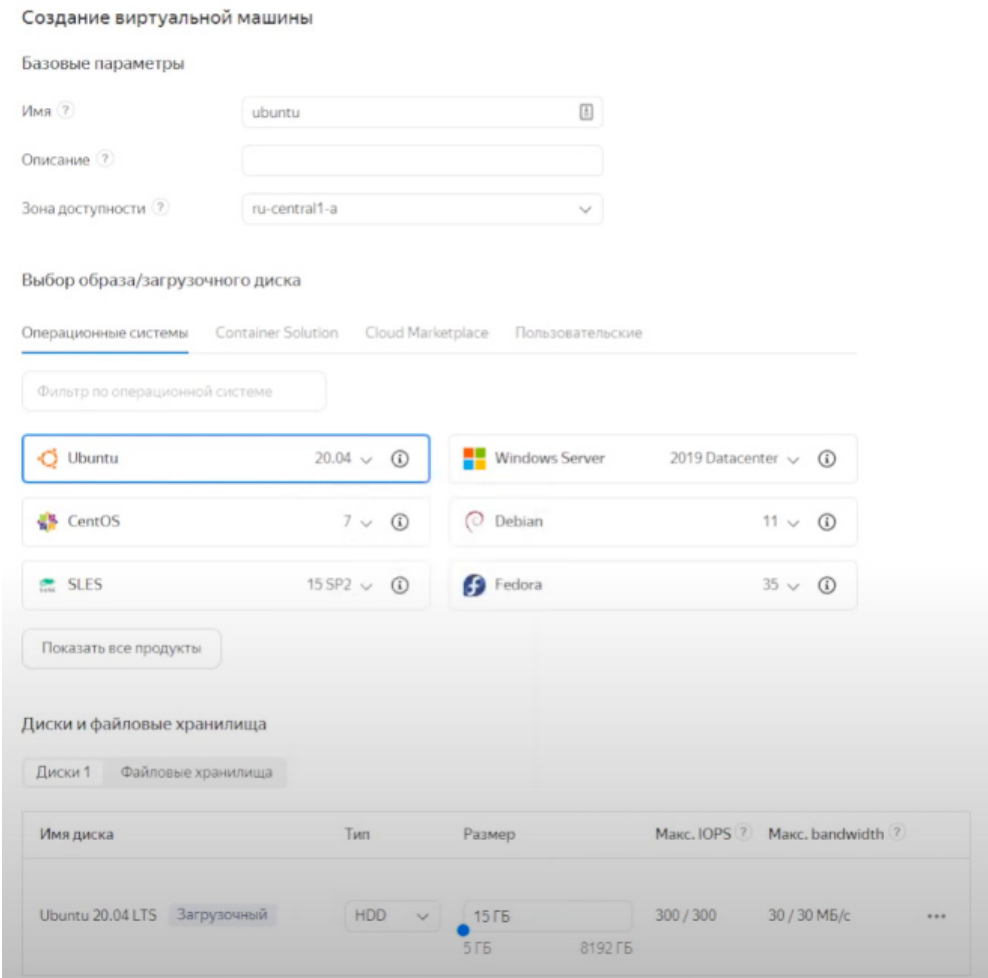
Выполнение работы:

Теоретический блок:

Yandex Compute Cloud позволяет использовать виртуальные машины в инфраструктуре Yandex для решения различных задач. Можно разместить в Compute Cloud своё готовое приложение или инфраструктуру для разработки, провести нагрузочное и функциональное тестирование.

Практический блок

Создадим виртуальную машину, выбрав необходимые параметры, такие как имя, ОС, диски и прочее.



Создание виртуальной машины

Базовые параметры

Имя ? ubuntu ⓘ

Описание ?

Зона доступности ? ru-central1-a ▼

Выбор образа/загрузочного диска

Операционные системы Container Solution Cloud Marketplace Пользовательские

Фильтр по операционной системе

Ubuntu 20.04 ⓘ	Windows Server 2019 Datacenter ⓘ
CentOS 7 ⓘ	Debian 11 ⓘ
SLES 15 SP2 ⓘ	Fedora 35 ⓘ

Показать все продукты

Диски и файловые хранилища

Диски 1 Файловые хранилища

Имя диска	Тип	Размер	Макс. IOPS ?	Макс. bandwidth ?	
Ubuntu 20.04 LTS	Загрузочный HDD	15 ГБ	300 / 300	30 / 30 МБ/с	...
		5 ГБ	8192 ГБ		

Рисунок 1 – Создание VM

Диски и файловые хранилища

Диски 1 Файловые хранилища

Имя диска	Тип	Размер	Макс. IOPS ?	Макс. bandwidth ?
Ubuntu 20.04 LTS	Загрузочный	HDD	300 / 300	30 / 30 МБ/с
		15 ГБ		
		5 ГБ		
		8192 ГБ		

Добавить диск

Вычислительные ресурсы

Платформа ? Intel Ice Lake

vCPU 2 / 2 - 96

Гарантированная доля vCPU ? 20% 50% 100%

Для тестирования прототипов без нагрузки или с минимальной нагрузкой.

RAM 2 ГБ / 1 ГБ - 8 ГБ

Дополнительно ☐ Прерываемая ?

Сетевые настройки

Подсеть ? default / default-a

Публичный адрес Автоматически Список Без адреса

Дополнительно ☐ Защита от DDoS-атак ?

Рисунок 2 – Создание VM. Выбор вычислительных ресурсов

В разделе “Доступ” сгенерируем SSH-ключ для созданной VM.

```

mRemoteNG - confCons.xml - local_ubuntu
Подключение: SSH2
local_ubuntu
Using username "ubuntu".
Welcome to Ubuntu 22.04 LTS (GNU/Linux 5.15.0-33-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Sun May 29 05:37:14 PM UTC 2022

System load:  0.087890625   Processes:           143
Usage of /:   51.6% of 18.93GB Users logged in:       0
Memory usage: 7%           IPv4 address for docker0: 172.17.0.1
Swap usage:   0%           IPv4 address for enp0s3: 192.168.31.43

 * Super-optimized for small spaces - read how we shrank the memory
 * footprint of MicroK8s to make it the smallest full K8s around.
 * https://ubuntu.com/blog/microk8s-memory-optimisation

20 updates can be applied immediately.
2 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Last login: Sun May 29 13:22:20 2022 from 192.168.31.207
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa): ^C
ubuntu@ubuntu:~$ cat /home/ubuntu/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQBAQCnScqbtz6RRKQuGxdRKOe/CBTLeSBDSelPhDMTfWJjpuY9BDG8tvmKevpdr
7cZwU3fINJE+Ja8Mpaalzk9ghwcbRJCj1W5SjCOga4prQBvCnS3xHR4Uz1TBYsqyzqBG9wcL0ryzJfCHKYQlqma4QKZ3RqdU8
H6krkpnzndcWzrgYf0TwrFLFhhNB33fRYb0vuPrMnpRA2yY7o1AqBzk305HbyY+gQ2WRpyULCgSdlqtMvTsiG0Qn7d7zEvzOM9
ctZBM2iXLC1NVnueXGJQcaAyW/q+dy/B8crspTNfBeBn85sfurITGgdkyFiyvCnywagYtEm3P3Jo8COW5e9GVu81/dsCYeYw
1/Kgc6BardBWgir521i9GuzN2Rwm0baIK4/yzV68D+qSo7nVOEb7qvEFBt7LX8htUik2TFVv2GbmURmKAy43CoqFIqV3nmCc5
1xkF7EBGqjN3wQp37v7f1oauJm2Y34QrWziMXQ3o/tipTR4RKA2S/aBAfd3CteUiEXR/W+0= ubuntu@ubuntu
ubuntu@ubuntu:~$
  
```

Рисунок 3 – Генерация SSH-ключа

После создания VM, в разделе Виртуальные машины появится созданная виртуальная машина:

Виртуальные машины

Фильтр по имени: Все статусы: Все зоны доступности:

Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Привязанная	Размер дисков	Зона доступности	Внутренний IPv4	Публичный IPv4	Дата создания	Идентификатор
ubuntu	Running		Intel Ice Lake	2	20%	2 Гб	да	15 Гб	ru-central1-a	10.10.0.34	51.250.93.72	29 мая 2022, в 20:49	f7e7f7441e0c15b8f2

Рисунок 4 – Созданная VM

После этого, можно подключиться к созданной VM по ssh:

```

Last login: Sun May 29 13:22:20 2022 from 192.168.31.207
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ubuntu/.ssh/id_rsa): ^C
ubuntu@ubuntu:~$ cat /home/ubuntu/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCgQCNsCqbtzGRKQuGxdRKOo/CBTLeaBDSe1PhDMTfWTjpuY9BDOGmVmkKevpdn
7tc2Wu3fNJE+Ja8MpsalzK9ghwcbRJCj1W5SjCOgs4pvQBvCnSjXHR4Uz1lTBYsqyzqBG9wcl0ryzJfCKYQqlqma4QK23KqdUb
H6krkpzndcWargYo7RwFLFhhNB53fAYb0vuPrMnpRA2yY7olAq8zk30SHbyY+qQ2WRpyULCgGd1qtMvTs1G0Qn7d7zEvzQM9
tcZBM21XLC1NVnueXGJQcaAyW/q+dy/B8crspTNfBeBn8SsfurITGgdkyFiyvCnywagYtEm3P3Jo8COw5a9GVu81/dsCYaYw
l/Kg6BArdBWgir52l19GuzN2Kwm0baIK4/yzV68D+qSo7nVOEb7qvEFBt7LX8htUik2TFVvZGbmURmKAY43CoqFIqV3nmCc5
lxF7EhGBqJN3wQpj7v7floauJm2Y34QrWziMXQ3o/tipTR4RKAAS/aBAfd3CtsUiRXR/W+0= ubuntu@ubuntu
ubuntu@ubuntu:~$ ^C
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$
ubuntu@ubuntu:~$ ssh ubuntu@51.250.93.72
The authenticity of host '51.250.93.72 (51.250.93.72)' can't be established.
ED25519 key fingerprint is SHA256:nBgFS2IfkKjEn3ejf3xdK2+57SjzdMfVxbzq//WrMqk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '51.250.93.72' (ED25519) to the list of known hosts.
Enter passphrase for key '/home/ubuntu/.ssh/id_rsa':
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-110-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

ubuntu@ubuntu:~$
  
```

Рисунок 5 – Подключение к VM

В случае необходимости, динамический по умолчанию ip адрес можно сделать статическим. Для этого нужно перейти в раздел Virtual Private Cloud, затем в подраздел IP-адреса, где адрес можно изменить на статический.

IP-адреса

Фильтровать по адресу:

IP-адрес	Тип	Зона доступности	Версия	Вид	Защита от DDoS-атак	Ресурс	Дата создания
51.250.93.72	Используется	Публичный	ru-central1-a	IPv4	Динамический	ubuntu Виртуальная машина	29 мая 2022, в 20:49

Переместить
Сделать статическим

Рисунок 6 – Установка статического IP-адреса на ВМ

Если нужно создать несколько ВМ, которые требуют настройки, можно использовать утилиты автоматизации (Terraform), помогающие развернуть инфраструктуру в облаке. Описав через скрипт необходимую инфраструктуру, впоследствии при помощи единственной команды можно развернуть её в облаке.

```
sudo mkdir -p /etc/containerd
containerd config default | sudo tee /etc/containerd/config.toml
sudo systemctl restart containerd
sudo apt-get install -y apt-transport-https ca-certificates curl
sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg https://packages.cloud.google.com/apt/doc/apt-key.gpg
echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg] https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /etc/apt/sources.list.d/kubernetes.list
sudo apt-get update
sudo apt-get install -y kubelet kubeadm kubectl
sudo apt-mark hold kubelet kubeadm kubectl
path: /home/yc-user/install.sh
permissions: '0777'
runcmd:
- [ bash, /home/yc-user/install.sh ]
ubuntu@ubuntu:~/Creat-3-VMs$
ubuntu@ubuntu:~/Creat-3-VMs$
ubuntu@ubuntu:~/Creat-3-VMs$
ubuntu@ubuntu:~/Creat-3-VMs$
ubuntu@ubuntu:~/Creat-3-VMs$ ls
cloud-config-master.yaml  create-k8s-vm.tf  terraform.tfstate.backup
cloud-config-worker.yaml  terraform.tfstate
ubuntu@ubuntu:~/Creat-3-VMs$ terraform init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of yandex-cloud/yandex from the dependency lock file
- Using previously-installed yandex-cloud/yandex v0.73.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ubuntu:~/Creat-3-VMs$ terraform
```

Рисунок 7 – Проверка доступности Terraform на Yandex Compute Cloud

В результате появляются новые ВМ, к которым также можно подключиться по ssh.

Виртуальные машины

Фильтр по имени

Все статусы

Все зоны доступности

☐	Имя	Статус	ОС	Платформа	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности	Внутренний IPv4	Публичный IPv4	Дата создания	Идентификатор	
☐	worker2	Running		Intel Broadwell	4	100 %	16 Гб	да	50 Гб	ru-central1-b	10.3.0.12	51.250.29.58	29 мая 2022, в 20:55	epdohhecpuhta0aqq3m	...
☐	worker1	Running		Intel Broadwell	4	100 %	16 Гб	да	50 Гб	ru-central1-a	10.2.0.11	51.250.92.129	29 мая 2022, в 20:55	fhwitf7rehtovs7pasq2	...
☐	master1	Running		Intel Broadwell	4	100 %	8 Гб	да	50 Гб	ru-central1-a	10.2.0.10	51.250.77.166	29 мая 2022, в 20:55	fhwk73ga3f7ep2raus59	...

Рисунок 8 – Созданные ВМ

Также при помощи Terraform можно удалить созданные виртуальные машины, используя на необходимой ВМ команду **terraform destroy**.

```

- resource "yandex_vpc_subnet" "subnet-1" {
  - created_at      = "2022-05-29T17:55:51Z" -> null
  - folder_id      = "blgb5itj5t2k5d30oi2c" -> null
  - id             = "e9bqla39g4ap2fknh2su" -> null
  - labels         = {} -> null
  - name           = "subnet1" -> null
  - network_id     = "enpla9n0vv40ui6j74iq" -> null
  - v4_cidr_blocks = [
    - "10.2.0.0/16",
  ] -> null
  - v6_cidr_blocks = [] -> null
  - zone           = "ru-central1-a" -> null
}

# yandex_vpc_subnet.subnet-2 will be destroyed
- resource "yandex_vpc_subnet" "subnet-2" {
  - created_at      = "2022-05-29T17:55:51Z" -> null
  - folder_id      = "blgb5itj5t2k5d30oi2c" -> null
  - id             = "e2l1blatgekqkhe65hg" -> null
  - labels         = {} -> null
  - name           = "subnet2" -> null
  - network_id     = "enpla9n0vv40ui6j74iq" -> null
  - v4_cidr_blocks = [
    - "10.3.0.0/16",
  ] -> null
  - v6_cidr_blocks = [] -> null
  - zone           = "ru-central1-b" -> null
}

```

Plan: 0 to add, 0 to change, 6 to destroy.

Do you really want to destroy all resources?

Terraform will destroy all your managed infrastructure, as shown above.
There is no undo. Only 'yes' will be accepted to confirm.

Enter a value: yes

```

yandex_compute_instance.vm-3: Destroying... [id=epdoh6hecphu8a0aqq3m]
yandex_compute_instance.vm-2: Destroying... [id=fhmitf7rehtova7pasq2]
yandex_compute_instance.vm-1: Destroying... [id=fhmk73ga3f7ep2raus59]

```

Рисунок 9 – Удаление созданных ВМ

Лабораторная работа №6

Тема: Подключение к облачной базе данных

Занятие 17. Практическая работа 9

Тема: Организация личного информационного пространства. Облачные хранилища данных

Цель выполнения практической работы: научиться создавать свой Яндекс. Диск; освоение технологии работы с сервисом Яндекса – «Яндекс. Диск».

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Облако — термин, под которым понимают пользование веб-сервисами, запущенными на удаленных серверах, которые принадлежат и предоставлены третьими лицами, к которым можно подключиться при помощи Интернета с любого устройства - будь то персональный компьютер, рабочий ноутбук, мобильный телефон или планшет.

Облачные технологии – это возможность иметь доступ к данным, не устанавливая специальных приложений на устройстве. Все необходимое обеспечение пользователям предоставляют серверы. Простыми словами, облачное хранилище, это сервис, который предоставляет для вас определенное место в интернете для хранения ваших файлов.

Вот один примеров использования виртуального облака:

Сейчас становится неактуально держать всю свою музыкальную коллекцию на локальном жестком диске. Например, облачный сервис, как "Яндекс музыка". Это очень удобно — иметь доступ к любой музыкальной композиции онлайн и располагать возможностью создания онлайн плейлистов.

Яндекс.Диск — бесплатный облачный сервис от Яндекса, позволяющий пользователям хранить свои данные на серверах в облаке и передавать их другим пользователям в интернете. Работа построена на синхронизации данных между различными устройствами. В настоящее время регистрация пользователей доступна всем. Ранее, до запуска Яндекс. Диска, функции хранения пользовательских файлов на Яндексе выполнял сервис Яндекс.Народ.

Изначально Яндекс.Диск предоставляет около 10 Гб навсегда.

Кроме того, Яндекс.Диск может выступать в качестве службы облачного сервиса, интегрируясь в офисный пакет Microsoft Office, а недавно появилась возможность автоматической загрузки фото и видеофайлов с цифровых камер и

внешних носителей информации на Яндекс. Диск. При этом пользователю предоставляются дополнительно 32 ГБ пространства на полгода.

Методы работы с Яндекс. Диск

Сервисом Яндекс.Диск можно пользоваться двумя способами:

- 1) Можно заходить в папку Яндекс.Диска по публичной ссылке (публичная ссылка – это ссылка на файлы или папки, предназначенные для общего доступа), отправленной вам преподавателем или другом, и пользоваться данными.
- 2) Можно создавать собственные ресурсы, личные или предназначенные для общего доступа в облаке, установив Яндекс.Диск на свой компьютер.

Яндекс Диск — использование через браузер

Для того чтобы не загружать приложение, экономя таким образом системные ресурсы вы сможете получить доступ к любой информации ранее загруженной на облако через браузер. Существуют несколько отличий в использовании этих подходов:

-Используя этот метод, вы не получаете синхронизации информации с Яндекс Диском;

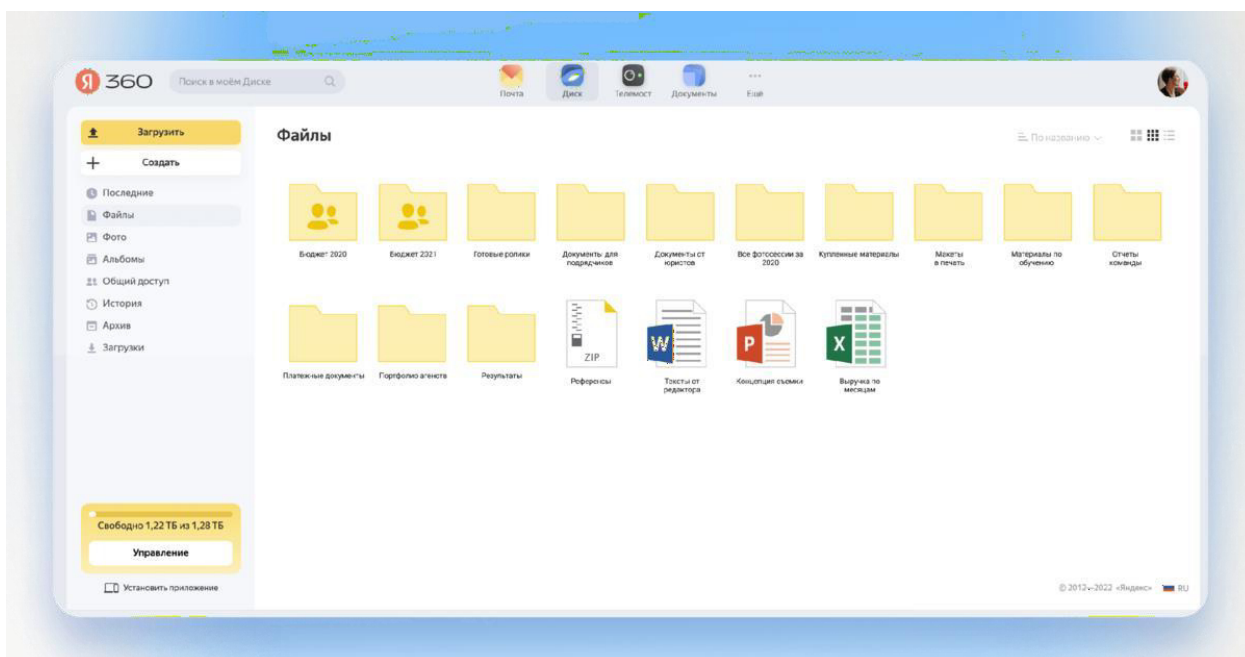
- Некоторые браузер накладывают ограничение на загрузку файлов с объёмом выше 2Гб, поэтому может случиться обрыв соединения;
- Предоставляется возможность пред просмотра файлов, не все типы содержимого поддерживаются, но текстовые файлы, вроде Word, изображения и подобные, можно предварительно изучить;
- В онлайн режиме существует ещё одна важная функция, а именно корзина, из которой можно достать ошибочно удалённые файлы;
- Удобный интерфейс с различными фильтрами по поиску необходимого содержимого.

ПРАКТИЧЕСКАЯ ЧАСТЬ

Задание 1. Создайте свой почтовый ящик на Яндексе (или войдите в него).

Как войти в Яндекс Диск

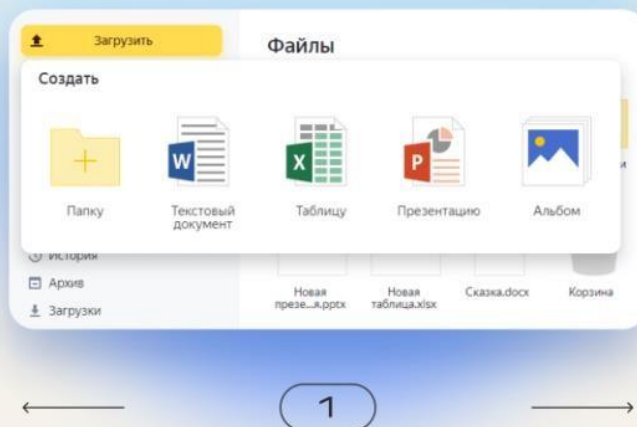
Если у вас есть аккаунт в Яндексе, авторизуйтесь и перейдите [по ссылке](#). Диск сразу откроется, можно начинать работу. Если учётной записи в Яндексе нет, сначала [зарегистрируйтесь](#).



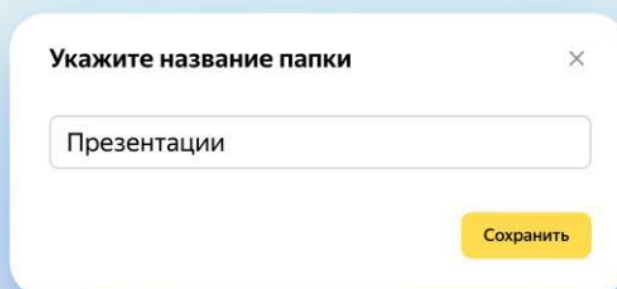
Так выглядит главная страница Диска. Слева — навигационное меню, в центре — ваши файлы и папки. Внизу информация, сколько свободного места осталось

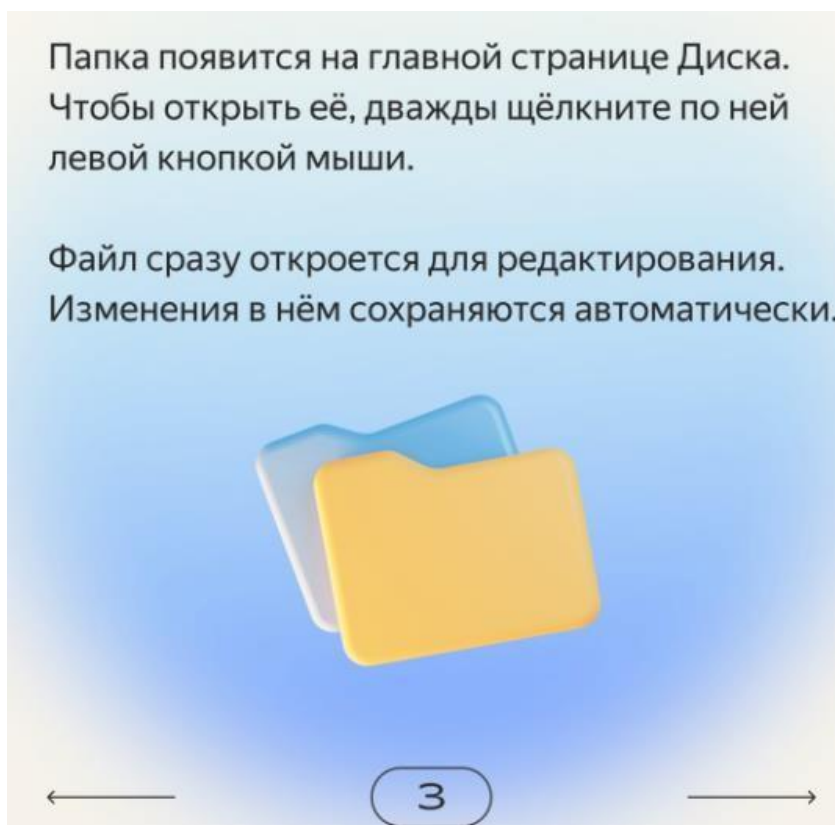
Задание 2. Выполните вариант создания собственной папки в облаке: Для этого:

Нажмите на кнопку **Создать**.
Откроется меню. В нём выберите нужный вариант: «Папку», «Текстовый документ», «Презентацию» или «Таблицу».



Придумайте название папке или файлу и нажмите **Сохранить**.



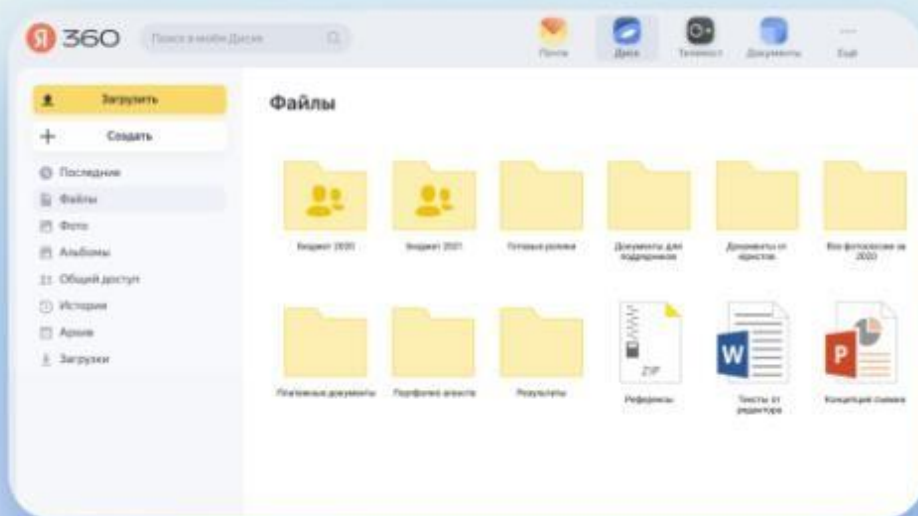


2. Загрузите на Яндекс.Диск любой файл из ваших файлов документов или рисунков.

Как загрузить файл на Диск

На Диск можно загружать файлы в любых форматах. Например, картинки в JPG или шаблоны в PSD, а ещё музыку, видео, документы и другие. Файлы в популярных форматах, таких как JPG, PPTX, DOCX, XLSX, открываются прямо на Диске. Не нужно их скачивать или устанавливать дополнительные программы, чтобы посмотреть.

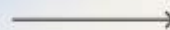
Откройте главную страницу Диска или папку на нём.



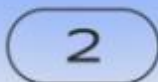
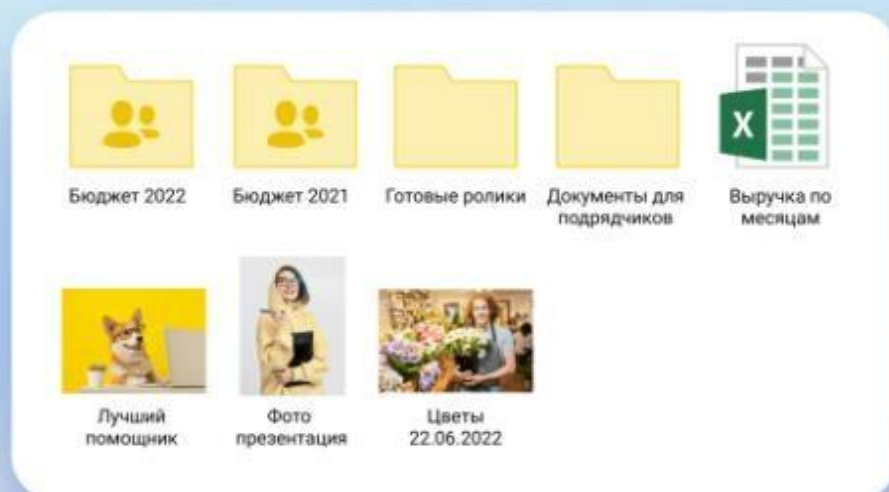
Максимальный размер файла для загрузки ограничен — от 1 до 50 ГБ в зависимости от тарифа.



1



Способ №1. Выберите нужные файлы на компьютере и мышкой перетащите их на страницу Диска. Загрузка начнётся автоматически.



Точно так же вы можете добавить на Диск папки с файлами, а не каждый по отдельности.



Фото
корпоратив



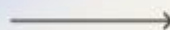
Отпуск
лето 2022



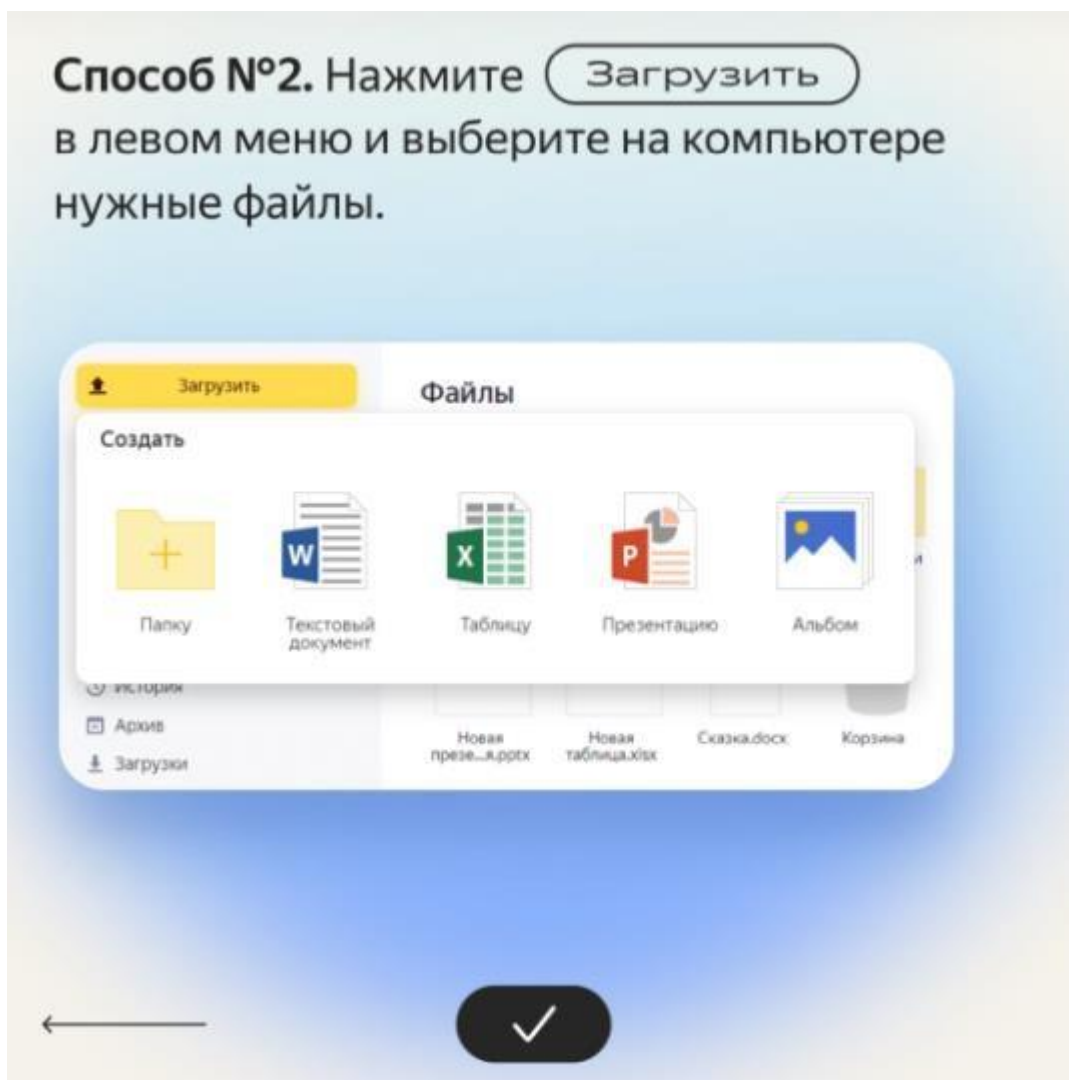
Разные
фотки



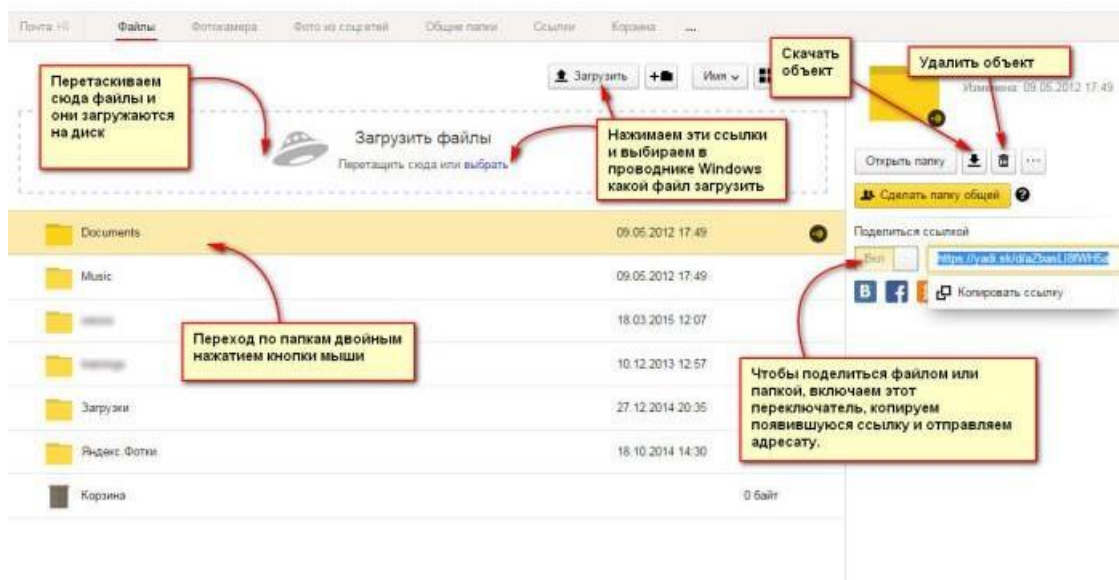
3



Способ №2. Нажмите **Загрузить** в левом меню и выберите на компьютере нужные файлы.



3. Загружаем файл.
4. Для операций над файлом или папкой, необходимо выделить их. После выделения справа появится контекстное меню с возможными действиями. Вы можете:
 - Скачать файл или папку
 - Удалить
 - Поделиться (предоставить доступ к файлу/папке другому пользователю скопировав ссылку).
 - Поделиться в социальных сетях.



12. Правой кнопкой мыши на значке вашего файла(который вы создадите в следующем задании) вызовите контекстное меню, выберите команду «поделиться», скопируйте появившуюся ссылку и отправьте ее преподавателю.

Задание 3. Работа с Яндекс.Диском

1. На **Рабочем столе/в телефоне** создать папку с именем **ПР8**, в папке создать **Документ Microsoft Word** с именем **Отчет_Практическая работа № 8**
2. Войдите в аккаунт на сервисе ЯндексДиск. (сделайте скриншот и поместите его в документ **Отчет_Практическая работа № 8**).
3. Создайте папку и загрузите на Яндекс.Диск любой файл из ваших файлов документов или рисунков. (сделайте скриншот и поместите его в документ **Отчет_Практическая работа № 8**).
4. Поделитесь ссылкой на загруженный файл с одноклассниками с помощью эл. почты. (сделайте скриншот и поместите его в документ **Отчет_Практическая работа № 8**).
5. Ознакомьтесь с Яндекс сервисами. (сделайте скриншот и поместите его в документ **Отчет_Практическая работа № 8**).
6. Попробуйте специальный поиск на сервисе Яндекс. (сделайте скриншот и поместите его в документ **Отчет_Практическая работа № 8**).
7. Перейдите в сервис Яндекс.Новости (узнайте актуальную информацию в разделе политика) (сделайте скриншот и поместите его в документ **Отчет_Практическая работа № 8**).
8. Сохранить документ **Отчет_Практическая работа № 8**.

□

9. Поделитесь документом *Отчет_Практическая_работа № 8 с преподавателем*. Выйти из своего аккаунта, закрыть браузер.

Контрольные вопросы

1. Дайте определение облачные технологии.
2. Дайте определение Яндекс.Диск.
3. Назовите методы работы с Яндекс.Диск.
4. Назовите какие еще облачные хранилища вам знакомы.
5. Для чего необходимы облачные хранилища?

Лабораторная работа №7

Тема: Создание задачи и доски в Trello

Цель: Создание презентации. Использование различных возможностей компьютерной презентации с помощью программы Power Point.

Оборудование: Программное обеспечение MS Windows.

Создание первого проекта

В Trello можно работать достаточно гибко и быстро, чтобы управлять проектом любого масштаба от начала и до конца. Узнайте, как создать свою первую доску и приступить к работе.

Создайте свой первый проект в Trello

Как и в любом другом проекте, вы будете управлять большим объемом информации, отслеживать статусы заданий и привлекать нужные заинтересованные стороны на каждом этапе работы. Ваша задача — продвигаться вперед в устойчивом темпе, зная, что результат будет поставлен в срок и в рамках бюджета, и сохранять улыбку на лице.

С Trello у вас будет поддержка на каждом этапе этого пути. Теперь, когда вы узнали основное о настройке доски Trello, самое время приступить к работе над первым проектом.

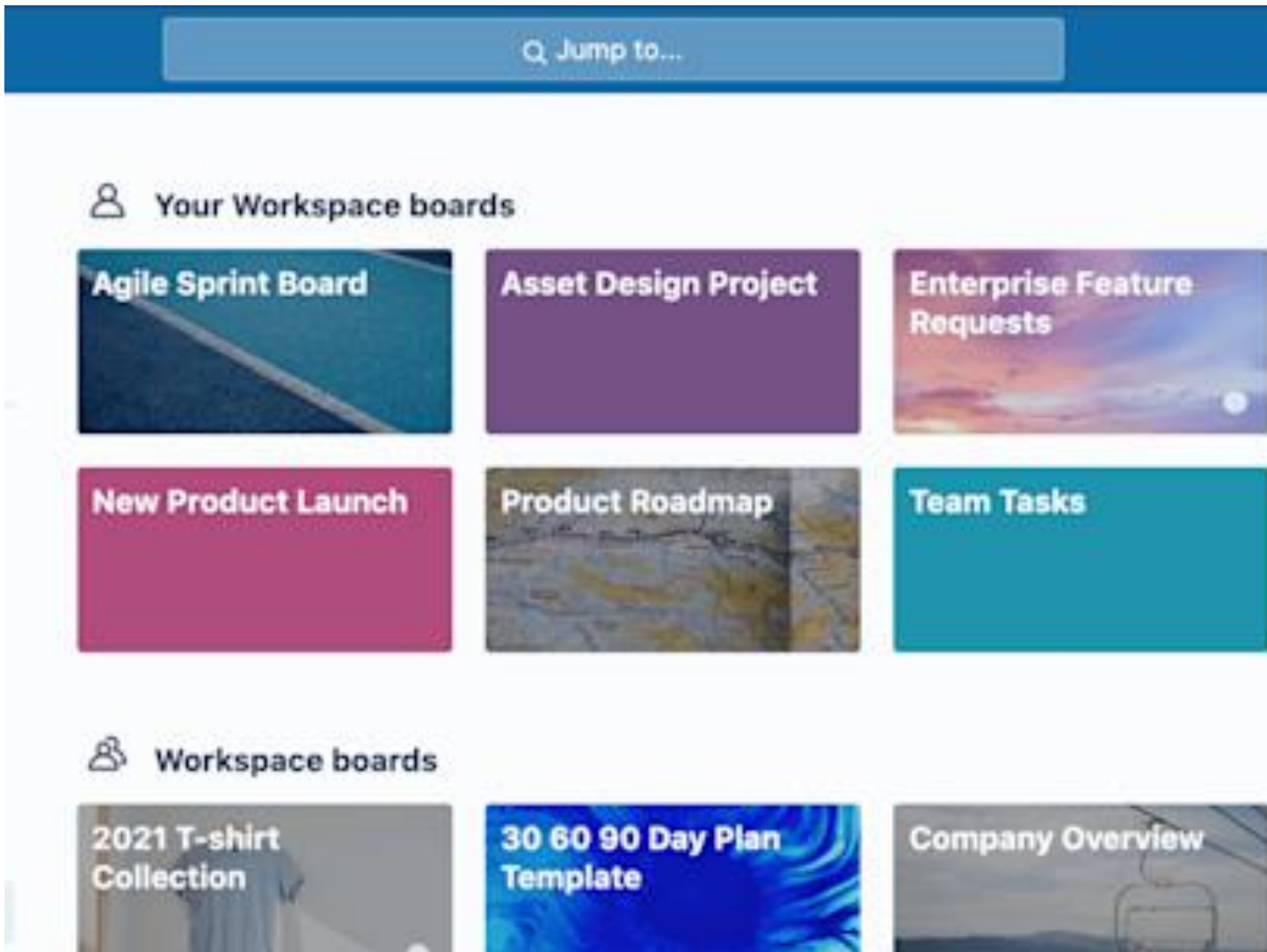
Обзор раздела

Из этого раздела вы узнаете следующее.

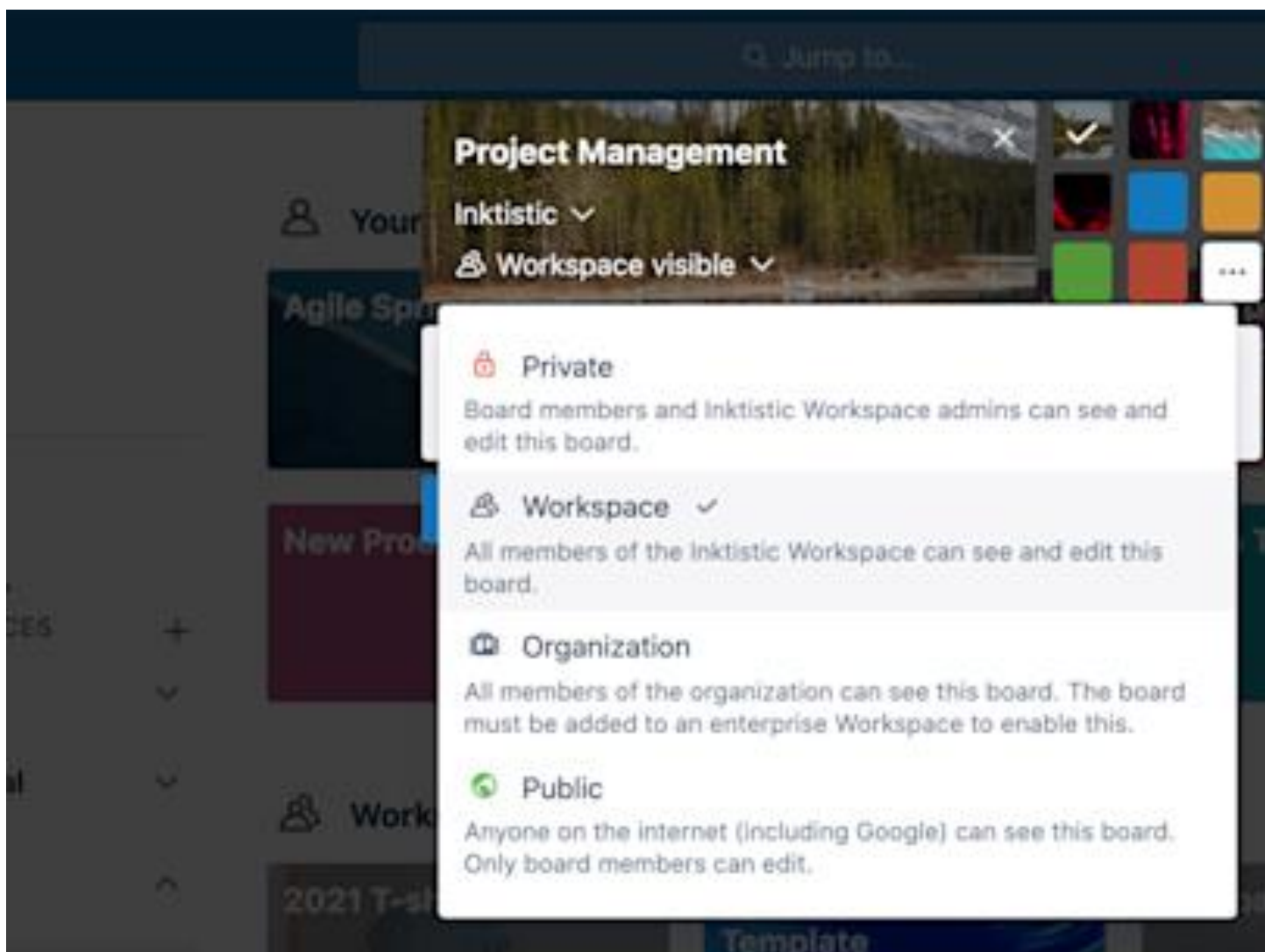
- Создание доски
- Настройка доски
- Начало совместной работы
- Создание рабочего процесса
- Добавление заданий и задач

Создание доски

Учитывая количество участников, идей, заданий и дат выполнения, входящих в проект, не так-то просто разработать оптимальную структуру для совместной работы. К счастью, вы можете начать с чистого листа и использовать доску Trello. Она поможет вам организовать все детали и добиться успеха.



- На вкладке Workspace (Рабочее пространство) вашей команды нажмите Create new board (Создать новую доску) либо нажмите значок плюса (+) на верхней панели Trello слева от вашего имени и выберите Create Board (Создать доску).
- Дайте доске название, чтобы пояснить, над чем вы работаете, от организации мероприятия или управления блогом до отслеживания потенциальных клиентов или запуска нового продукта.



Данные проекта и связанная с ним информация — ваши самые ценные ресурсы. При создании новой доски Trello у вас будет возможность настроить уровень конфиденциальности.

- При создании доски используется настройка по умолчанию **Workspace Visible** (Для рабочего пространства). Это означает, что просматривать, редактировать карточки и присоединяться к ним может любой участник данного рабочего пространства.
- Если изменить значение этой настройки на **Private** (Приватная), доска будет видна только ее непосредственным участникам. Как правило, рекомендуется устанавливать значение **Private** (Приватная) в качестве значения по умолчанию, если вы работаете над закрытым проектом, доступ к которому нужно ограничить для других участников команды. Этот параметр отлично подходит для личных досок, досок с персональными заданиями или досок, используемых совместно с клиентами.
- Доски типа **Public** (Публичная) могут просматривать все, у кого есть соответствующая ссылка. Публичные доски можно искать с помощью поисковых систем, например Google, а потому любая информация с такой доски доступна всем пользователям в Интернете. Эту настройку можно

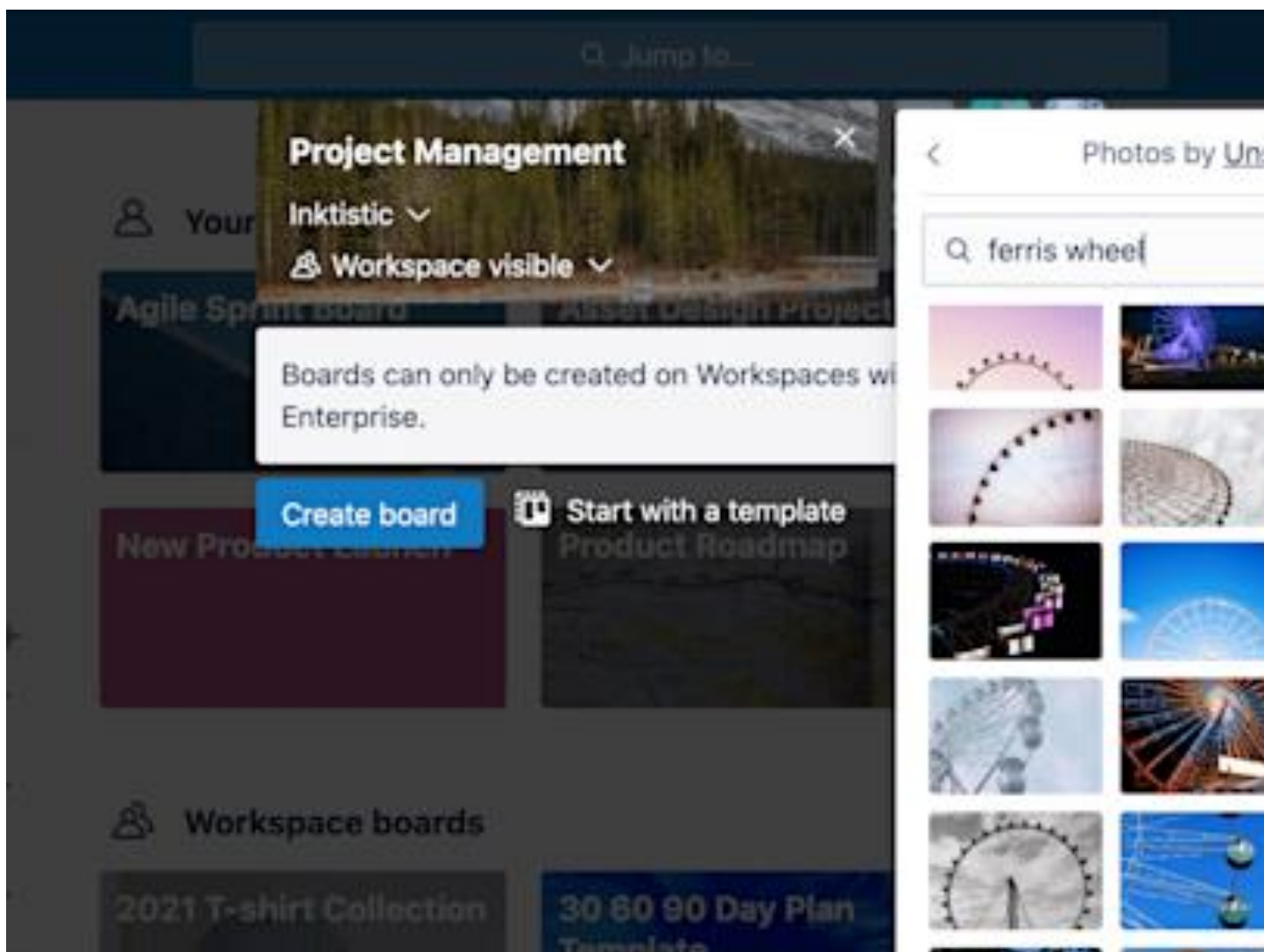
□

использовать для открытого проекта или дорожной карты продукта, в отношении которого нужно изучить общественное мнение. Наряду с другими параметрами доски, вы можете точно задать, кто может оставлять комментарии, добавлять или удалять участников, а также редактировать доску и присоединяться к ней.

Подробнее о публичных и частных досках см. [здесь](#).

Настройка доски

Не бывает двух одинаковых рабочих процессов — так почему бы не придать каждой доске свой уникальный внешний вид? Мы за гибкий и увлекательный формат работы, поэтому в Trello есть возможность настроить фон доски, чтобы отразить уникальный характер команды.

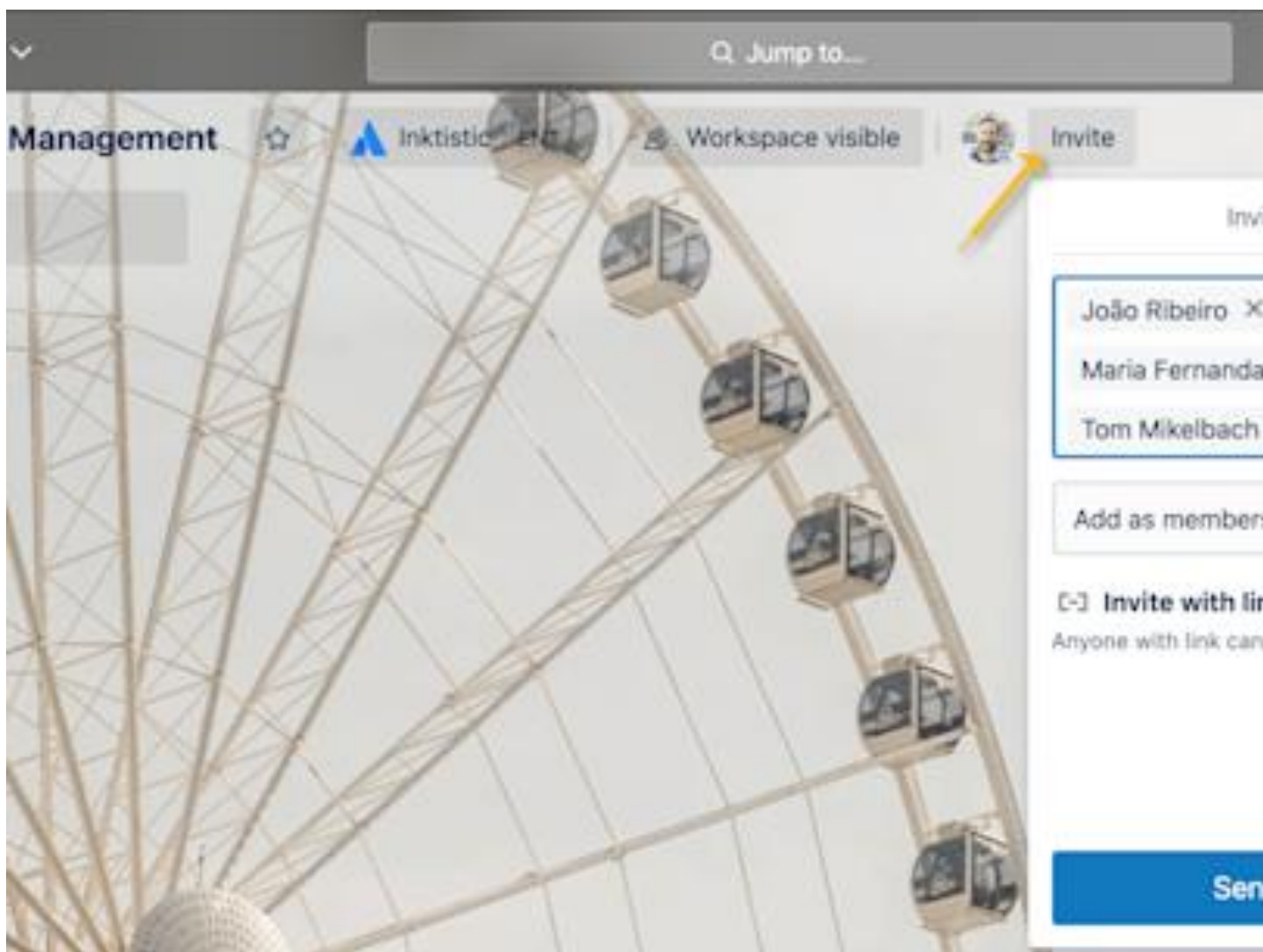


Создайте красивый и неповторимый фон доски: выберите один из основных цветов, загрузите собственное изображение* или найдите подходящее в огромной библиотеке фотографий [Unsplash](#). Изменить фон можно в любое время в меню доски.

Начало совместной работы

Совместная работа над проектом идет лучше, если все участники команды остаются в курсе происходящего — и неважно, руководит участник проектом или просто заглядывает в него периодически, чтобы проверить статус выполнения.

Пригласите участников на доску, чтобы назначить им задания и предоставить возможность совместной работы.



1. В верхней части доски нажмите Invite (Пригласить), затем выберите участника команды, которого хотите добавить на доску. Вы также можете найти и пригласить участника по адресу электронной почты или имени.
2. В нижней части меню Invite (Пригласить) доступна ссылка на доску, которой можно без труда поделиться со всеми нужными участниками. Опубликуйте ссылку в канале Slack или отправьте по электронной почте внешним заинтересованным сторонам, чтобы начать совместную работу. Кроме того, можно пригласить участников, загрузив сканируемый QR-код.

Настройка прав доступа участников

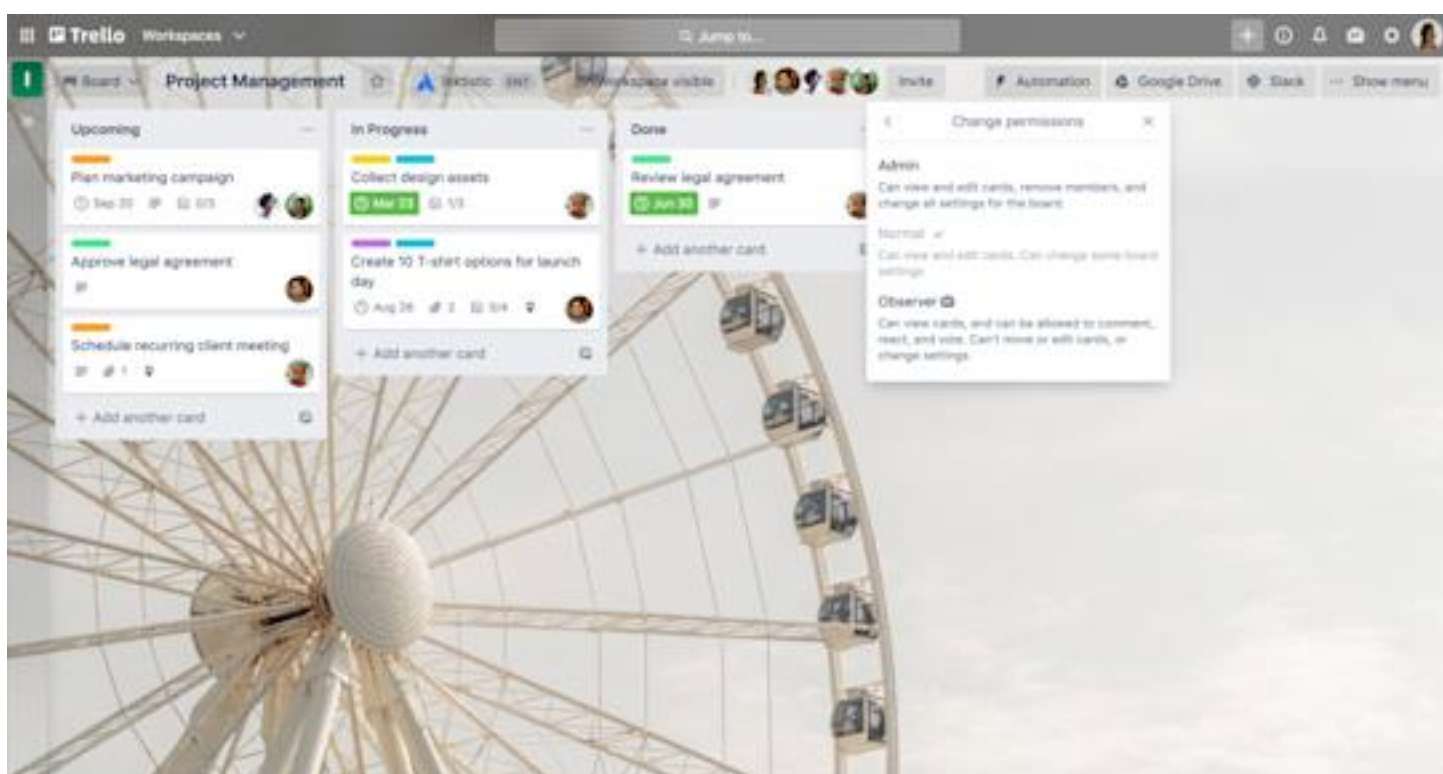
□

Настройте права доступа участников, чтобы четко определить их роли. Права доступа позволяют определить точный уровень доступа для конкретных сотрудников. Ознакомьтесь с приведенными ниже настройками пользователей, чтобы нужные сотрудники могли контролировать нужный объем информации.

Обычный пользователь. Может просматривать и редактировать карточки, а также, в зависимости от настроек, изменять некоторые права доступа к доскам. Отлично подходит для участников рабочего пространства, активно задействованных в проекте.

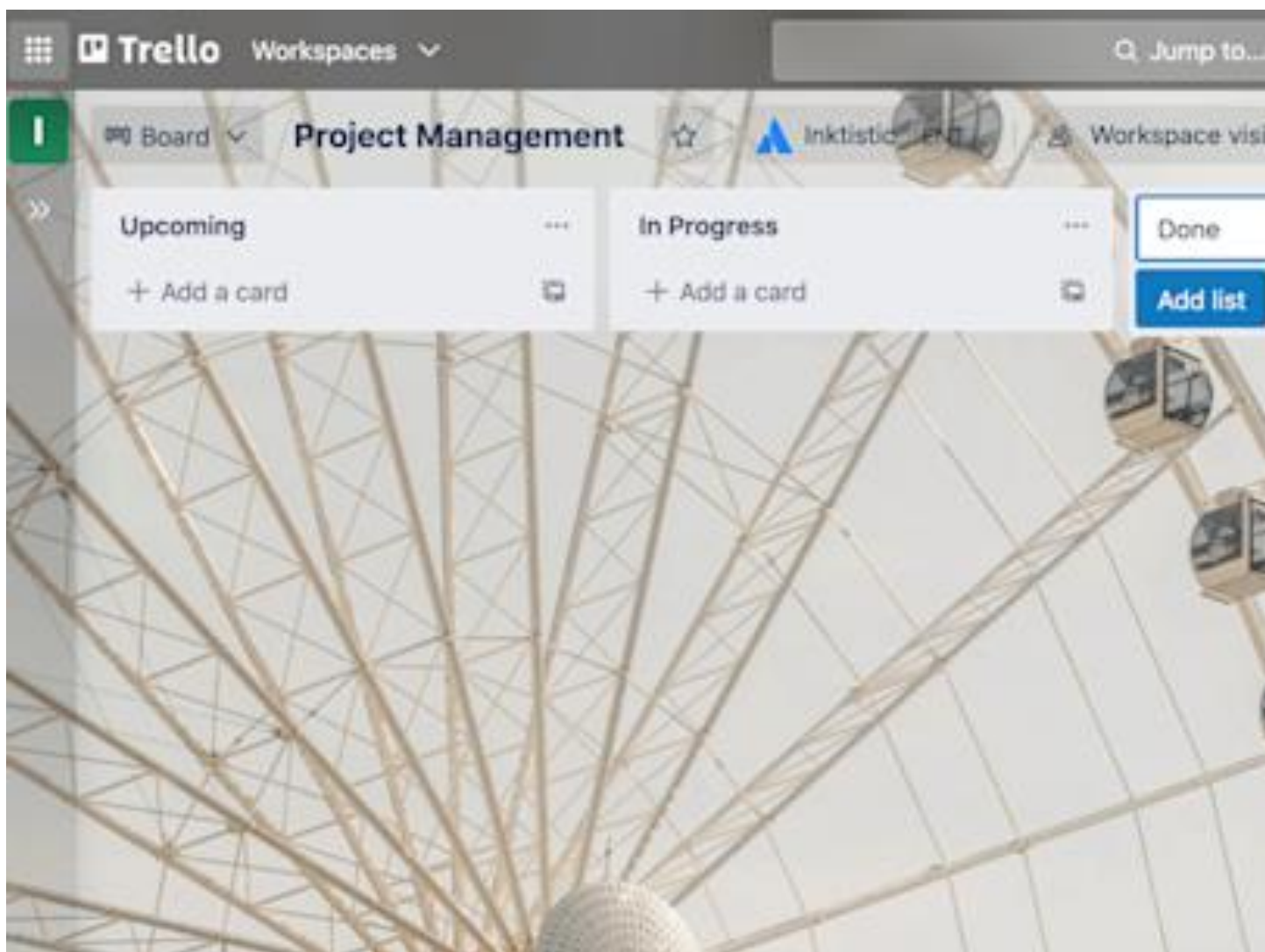
Администратор. Может просматривать и редактировать карточки, исключать участников и изменять все настройки доски. Отлично подходит для руководителей проектов и менеджеров, которые полностью контролируют проект.

Наблюдатель*. Может просматривать содержимое, отображаемое на доске, но не может редактировать карточки. Наблюдателям можно разрешить оставлять комментарии, отвечать с помощью эмодзи и выполнять другие действия в зависимости от настроек доски. Отлично подходит для досок с информацией для клиентов, менеджеров, третьих сторон и т. п.



Создание рабочего процесса

Колонки Trello позволяют упорядочить карточки по статусу. В колонке можно объединить ряд элементов, например идей или заданий, в рамках более крупного проекта. Но самая главная задача — будь то стандартная доска Kanban, конвейер продаж, маркетинговый календарь или сложная система управления проектами — сформировать рабочий процесс, который будет отражать стиль работы команды.



1. Нажмите Add a list (Добавить колонку), чтобы добавить на доску свою первую колонку. В названии отразите ее содержание или этап работы. Названия колонок могут быть простыми, как для этапов рабочего процесса («Нужно сделать», «В работе» или «Готово»), либо подробными, если это нужно для работы команды. Помните: Trello очень просто настроить под уникальные особенности вашего рабочего процесса, а колонкам можно давать любые названия.
2. Добавьте на доску столько колонок, сколько вам нужно для организации рабочего процесса.

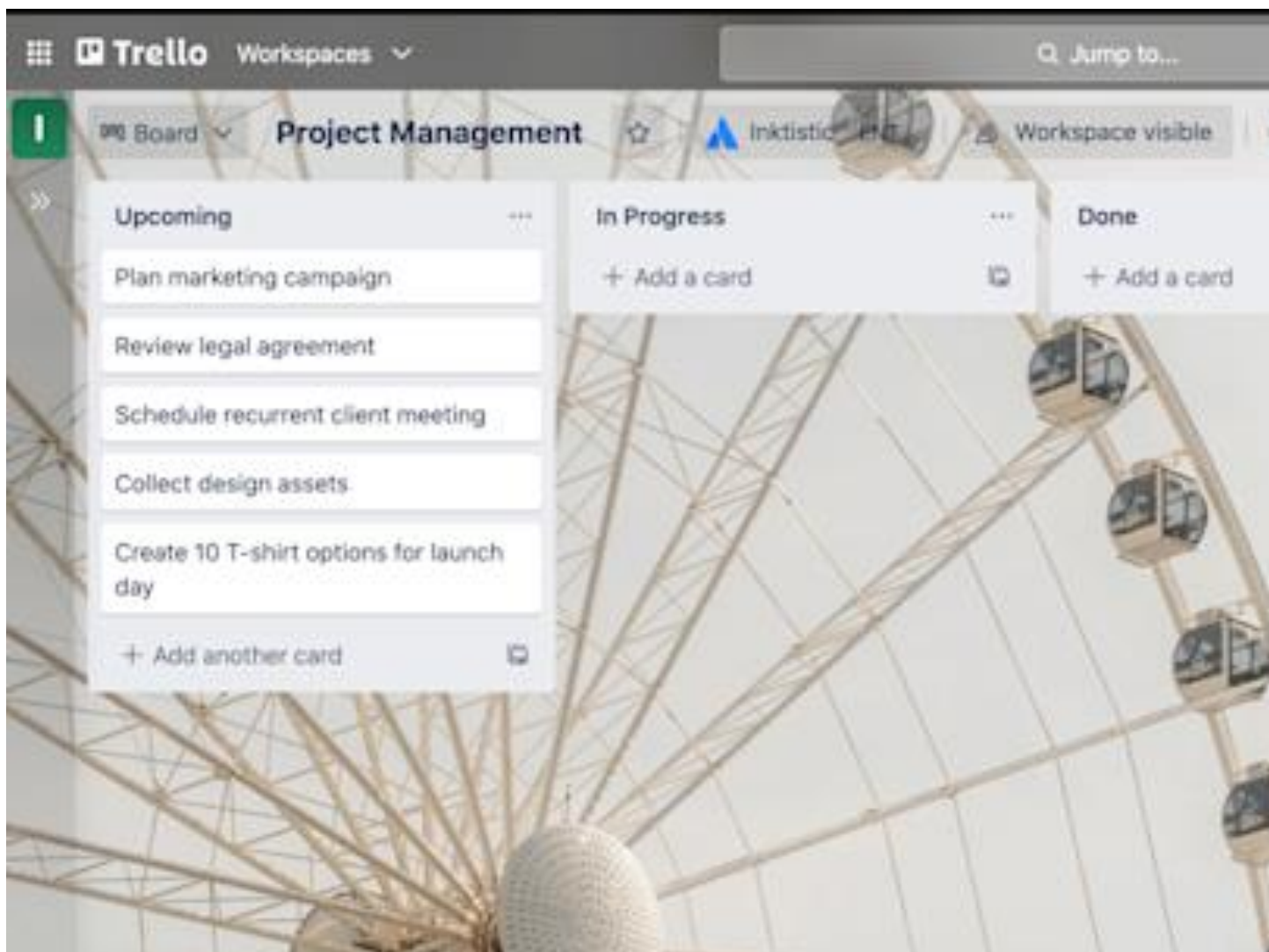
Добавление заданий и задач

Основной элемент доски, карточка, представляет собой задание или идею. Карточка описывает определенную задачу (например, задача по поддержке клиентов, разработка нового дизайна или оформление записи в блоге) либо содержит информацию, связанную с проектом, которая должна оставаться на виду.

Карточки можно настроить так, чтобы хранить самую разную полезную информацию, которую можно просмотреть одним нажатием. Перетаскивайте

□

карточки между колонками, чтобы видеть прогресс. Количество карточек на доске не ограничено.

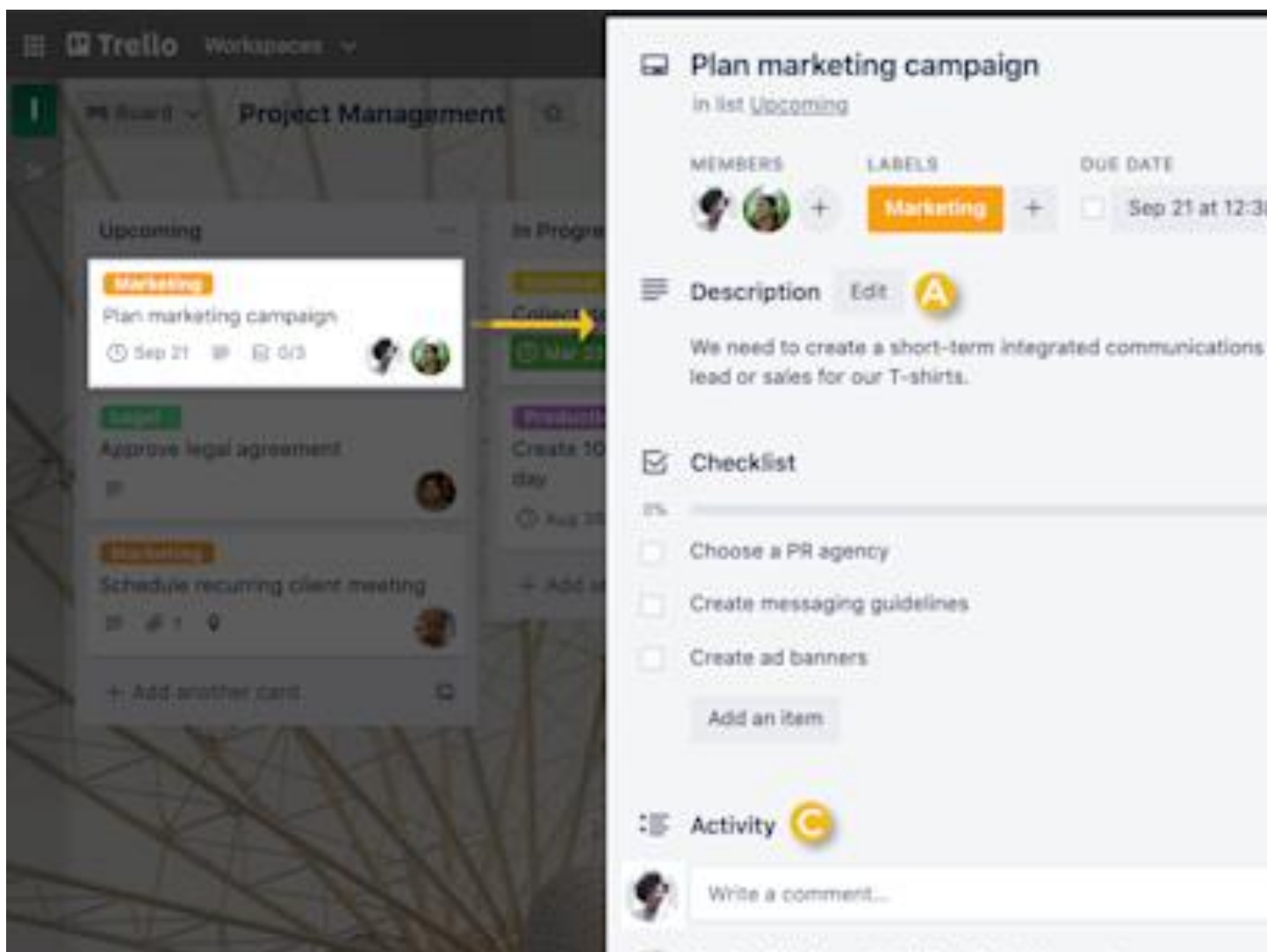


1. Добавляйте карточки для каждого задания, которое нужно выполнить, нажимая Add A Card (Добавить карточку) в первой колонке. Старайтесь давать карточкам короткие названия, чтобы их было проще найти на доске и соотнести со статусом.
2. Чтобы каждый сотрудник имел четкое представление о том, что нужно сделать, нажмите карточку и добавьте в нее дополнительную информацию, например следующую.
 - Описания
 - Даты выполнения
 - Участники
 - Списки задач
 - Вложения
 - Комментарии

Добавление подробностей на обратной стороне карточки

□

Как упоминалось ранее, нажав карточку, можно развернуть ее для просмотра или добавления сведений. Открывшееся окно называется обратной стороной карточки и содержит более подробную информацию.



Здесь находятся три основных раздела, с которыми стоит ознакомиться.

А. Описание карточки. При ежедневной коммуникации в команде важно пополнять контекст и представлять его в понятном виде. При наличии актуальной информации можно подготовиться к любому взаимодействию и избежать недопонимания в любых вопросах касательно проекта. Это особенно ценно для тех участников команды, которые работают с доской не ежедневно, а лишь периодически.

В поле описания можно добавить более подробную информацию о карточке, например ссылки на веб-сайты или пошаговые инструкции. Чтобы добавить сведения в карточку, нажмите Edit the description (Редактировать описание) в верхней части обратной стороны карточки. Здесь даже можно использовать разметку для форматирования текста.

В. Раздел добавления. В этом разделе обратной стороны карточки собраны дополнительные инструменты.

- Добавляйте участников в карточки, чтобы назначать им задания и без труда отслеживать, чем занят каждый и что еще требуется сделать.
- Добавляйте в карточки списки задач, которые содержат подзадачи или этапы выполнения. Так вы не упустите ни одной мелочи. Вы даже можете копировать списки задач из других карточек на доске. Кроме того, клиентам с тарифным планом [Standard](#) или [Premium](#) доступны продвинутые списки задач, для элементов которых можно назначать участников и даты выполнения, чтобы внести дополнительную ясность и подотчетность.
- Добавляйте в карточки даты, чтобы отслеживать сроки: все участники карточки получают уведомление за 24 часа до наступления срока. В меню дат можно выбрать дату начала и дату выполнения, чтобы отслеживать начало и окончание проектов и заданий. Как только задания будут выполнены, для соответствующих дат можно установить отметку о завершении.
- Добавляйте вложения с компьютера или из облачных хранилищ (например, Dropbox, Google Диск, Box или OneDrive).
- Обложки помогают дополнительно упорядочить доски и сделать их наглядными. Добавляя в карточки обложки и настраивая цвета, вы создаете визуальный контекст и можете использовать дополнительные обозначения и сведения. Вложенные изображения карточек можно отображать на их передней стороне как полноразмерные обложки. Параллельно с этим есть возможность менять цвета карточек.

Совет от профи

Точно настраивайте напоминания, выбирая для них конкретное время с привязкой к дате выполнения.

С. Комментарии и события. В ходе совместной работы и общения в команде можно добавлять комментарии к карточкам, в том числе обратную связь и обновления. Используйте в комментариях @упоминания участников доски или команды, чтобы отправить им уведомление в Trello. В ленте событий отображается хронология всех комментариев и действий с карточкой.

Лабораторная работа № 8

Тема: Настройка git hooks и lint-автоматизации

Что такое Prettier? Prettier – это средство для форматирования кода, которое нацелено на использование жёстко заданных правил по оформлению программ [1]. Оно форматирует код автоматически, обеспечивая соблюдение предустановленных правил форматирования кода для проекта, что в свою очередь позволяет сфокусироваться на более важных вещах при написании программного кода.

Инструмент Prettier обладает следующими возможностями и особенностями (рассмотрим основные из них).

– Приведение в порядок существующей кодовой базы можно выполнить одной командой. Ручная обработка больших объёмов кода займёт гораздо больше времени: представьте себе затраты труда, необходимые для ручного форматирования 20 000 строк кода.

– Prettier легко внедряется в проект, он использует наименее спорный подход к стилю при форматировании кода. Prettier – проект с открытым исходным кодом: многие внесли вклад в его доработку и улучшение.

– Prettier позволяет сосредоточиться на написании кода. Разработчики не осознают того, как много времени и сил тратится на форматирование кода. Использование Prettier в одном из наших проектов повысило эффективность работы на 10 %.

– Prettier помогает начинающим. Если вы начинающий программист, работающий в одной команде с серьёзными профессионалами, и вы хотите достойно смотреться на их фоне, в этом вам поможет Prettier.

– Затраты времени на code-review сокращаются в среднем на 15–20 %: нет необходимости напоминать о забытых пробелах, лишние запятых и других подобных ошибках – их исправит Prettier.

Для запуска локального форматирования кода в большинстве сред разработки предусмотрена возможность запуска форматирования текста прямо из контекстного меню, например как в WebStorm (рис. 1).

□

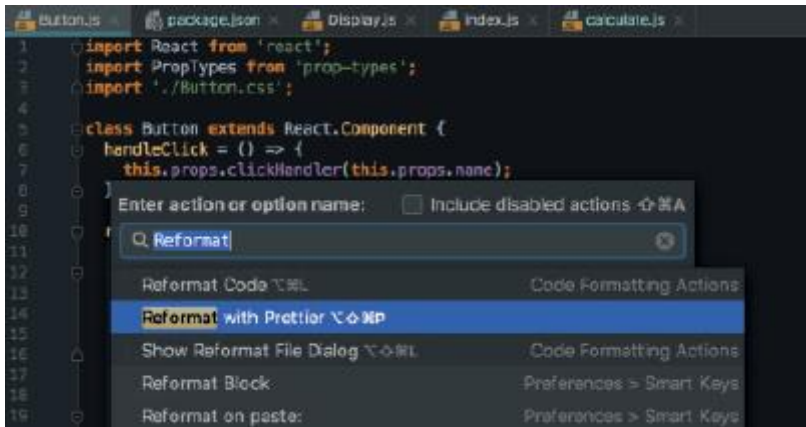


Рис. 1. Контекстное меню для запуска Prettier в IDE WebStorm

Что такое Git Hooks? У систем контроля версий есть механизм запуска callback-скриптов по определенному действию. Как и многие другие системы контроля версий, Git предоставляет возможность запуска пользовательских скриптов в случае возникновения определённых событий. Такие действия называются хуками и разделяются на две группы: серверные и клиентские. Если хуки на стороне клиента запускаются такими операциями, как слияние или создание коммита, то на стороне сервера они инициируются сетевыми операциями, такими как получение отправленного коммита. Хуки часто используются для широкого круга задач.

В некотором смысле хук является реализацией шаблона Observer из ООП. Хуки могут быть написаны на скриптовых языках, таких как Bash, Python, Ruby и других.

Рассмотрим виды git-hooks (рис. 2).

Для реализации нашей задачи мы будем использовать git-хук pre-commit для конфигурирования процесса отправки, проверки и форматирования изменений плагином, находящихся в коммите.

Цель исследования: исследование процесса подключения инструментов для автоматического форматирования программного кода при фиксировании изменений в git-репозитории путём коммита изменённых файлов.

Материалы и методы исследования

Далее приведено пошаговое руководство по установке и конфигурированию Prettier и локальных git-хуков в TypeScript-проекте.

1. Установка Prettier

Установить npm-пакет можно командой

```
npm install --save-dev prettier
```

2. Установка плагинов для Prettier

□

Установить npm-пакет плагина можно командой

```
npm install --save-dev prettier-plugin-organize-imports
```

Данный плагин необходим для автоматической сортировки импортов.

3. Установка плагинов для организации автоматической проверки и корректировки кода

Установить npm-пакеты можно командой

```
npm install --save-dev husky
```

Данный плагин позволяет настроить локальные хуки

```
npm install --save-dev lint-staged
```

Данный плагин позволяет настроить выполнение скриптов для файлов, готовых к коммиту.

4. Конфигурирование собственных правил исправления программного кода

Собственные правила в Prettier редактируются в корневом файле проекта.

Файл имеет название «.prettierrc». Ниже приведён пример конфигурации для соответствующего файла в Angular-проекте. Подробнее о конфигурировании правил Prettier можно прочесть на официальном сайте [2].

```
{  
  "bracketSpacing": true,  
  "printWidth": 140,  
  "semi": true,  
  "singleQuote": true,  
  "trailingComma": "all",  
  "tabWidth": 2,  
  "useTabs": false,  
  "overrides": [  
    {
```

□

```
"files": "src/**/*.ts",  
  
"options": {  
  
"parser": "typescript"  
  
}  
  
},  
  
{  
  
"files": "*.component.html",  
  
"options": {  
  
"parser": "angular"  
  
}  
  
},  
  
{  
  
"files": [".prettierrc", "package.json"],  
  
"options": {  
  
"parser": "json"  
  
}  
  
}  
  
]
```

5. Связывание Prettier с ESLint/TSlint

Данная статья подразумевает наличие установленного в проекте ESLint или TSLint. Для корректного подключения к проекту Prettier при наличии уже установленного инструмента проверки кода из вышеперечисленных можно воспользоваться инструкцией, расположенной на сайте [3].

6. Настройка локального pre-commit хука

□

Для этого в `package.json` необходимо в разделе `scripts` добавить скрипты для проверки и исправления текущего кода [4–6]:

```
"scripts": {  
  "prettier": "prettier --check \"src/**/*.ts|component.html\"\",  
  "prettier:fix": "npm run prettier -- --write"  
},
```

Затем в секции `husky` описать `pre-commit` хук:

```
"husky": {  
  "hooks": {  
    "pre-commit": "lint-staged"  
  }  
}
```

Далее описав скрипт для `lint-staged`:

```
"lint-staged": {  
  "**.(ts|component.html)": [  
    "npm run prettier:fix",  
    "git add"  
  ]  
}
```

7. Настройка универсальных параметров редактора в IDE

Для того чтобы все пользовались едиными настройками, такими как отступы или символы перевода строки, применяется файл `.editorconfig`. Он описывает единый набор правил в неоднородных командах для всех поддерживающих его IDE.

□

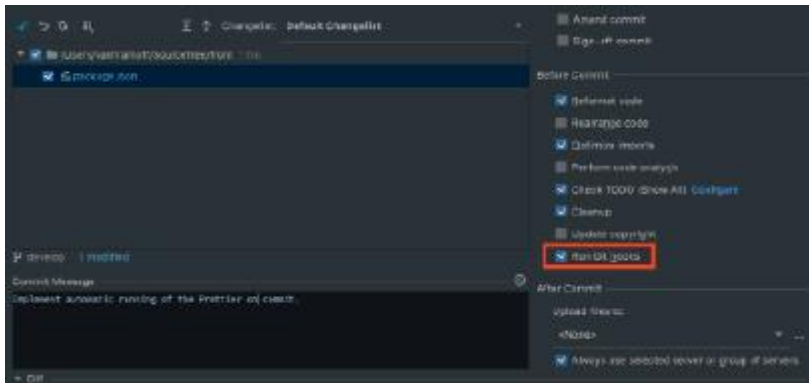


Рис. 3. Опция запуска Git-хуков в окне коммита в IDE WebStorm

На сайте проекта [7] можно найти список редакторов, которые поддерживают этот файл. В него входят WebStorm, AppCode, Atom, Eclipse, Emacs, VEdit и другие.

Для описания такого файла создадим в корне проекта файл «.editorconfig» и добавим в него следующий код:

Файл EditorConfig верхнего уровня

```
root = true
```

```
[*.md]
```

```
trim_trailing_whitespace = false
```

```
[*.js]
```

```
trim_trailing_whitespace = true
```

Переводы строк в стиле Unix с пустой строкой в конце файла

```
[*]
```

```
indent_style = space
```

```
indent_size = 2
```

```
end_of_line = lf
```

```
charset = utf-8
```

```
insert_final_newline = true
```

```
max_line_length = 100
```

8. Настройка автоматического исправления файлов, попавших в коммит

□

С этого момента IDE будет автоматически применять данный pre-commit хук. Главное – не снимать галочку, которая отвечает за выполнение подобных локальных хуков (рис. 2).

Результаты исследования и их обсуждение

В результате мы получили проект с настроенной системой автоматического форматирования кода. В проекте автоматически соблюдаются заранее предустановленные правила форматирования, что позволяет существенно экономить время на написание и ревю кода.

Выводы

Внедрение Prettier сократило время разработки проекта, так как не требовалось тратить дополнительное время на соблюдение кода-стиля для проекта. Автоматизация процесса приведения кода к стандартизированному форматированию через хуки позволила забыть про ручной контроль каждого измененного файла. Увеличилась читаемость кода: это стало заметно на программистах с опытом разработки менее 1 года, которые только пришли в проект.

Лабораторная работа № 9

Тема: Анализ утечек и проверка паролей

Цель работы

Получить практические навыки анализа Web-приложений на предмет наличия основных уязвимостей.

1. Теоретическая часть

На сегодняшний день по данным исследований, около 70% процентов сайтов содержат хотя бы одну уязвимость критического уровня. В компьютерной безопасности термин «уязвимость» (англ. *vulnerability*) используется для обозначения недостатка в системе, используя который можно намеренно нарушить её целостность и вызвать неправильную работу. Уязвимость может быть результатом ошибок программирования, недостатков, допущенных при проектировании системы, ненадежных паролей, вирусов и других вредоносных программ, скриптовых и SQL-инъекций. Некоторые уязвимости известны только теоретически, другие же активно используются и имеют известные эксплойты.

Обычно уязвимость позволяет атакующему «обмануть» приложение — заставить его совершить действие, на которое у того не должно быть прав. Это делается путём внедрения каким-либо образом в программу данных или кода в такие места, что программа воспримет их как «свои».

1.1. OWASP

Open Web Application Security Project (OWASP) — это открытый проект обеспечения безопасности веб-приложений. Сообщество OWASP включает в себя корпорации, образовательные организации и частных лиц со всего мира. Сообщество работает над созданием статей, учебных пособий, документации, инструментов и технологий, находящихся в свободном доступе.

Раз в несколько лет OWASP публикует рейтинг наиболее популярных уязвимостей в Web-приложениях, который называется OWASP Top 10. На данный момент список выглядит следующим образом:

- A1 Внедрение кода
- A2 Некорректная аутентификация и управление сессией
- A3 Межсайтовый скриптинг (XSS)
- A4 Нарушение контроля доступа
- A5 небезопасная конфигурация
- A6 Утечка чувствительных данных
- A7 Недостаточная защита от атак
- A8 Подделка межсайтовых запросов (CSRF)
- A9 Использование компонентов с известными уязвимостями
- A10 Незащищенный API

В рамках данной лабораторной работы будут разобраны некоторые из этих уязвимостей.

1.2. Внедрение кода (SQL Injection)

□

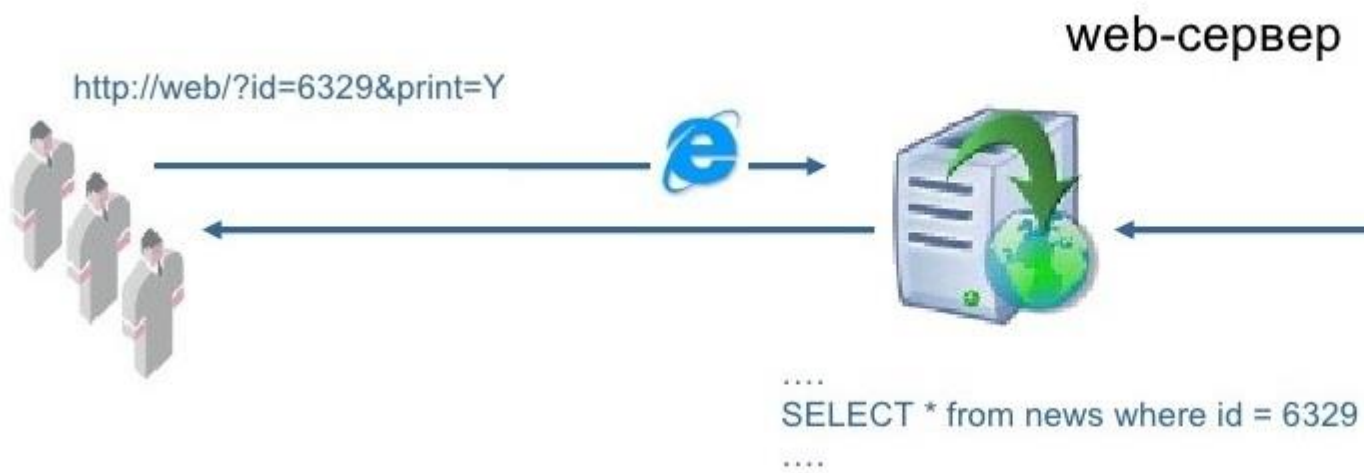
Все данные, как правило, хранятся в специальных базах, обращения к которым строятся в виде запросов, чаще всего написанных на специальном языке запросов SQL (*Structured Query Language* – структурированный язык запросов). Приложения используют SQL-запросы для того, чтобы получать, добавлять, изменять или удалять данные, например при редактировании пользователем своих личных данных или заполнении анкеты на сайте. При недостаточной проверке данных от пользователя, злоумышленник может внедрить в форму Web-интерфейса приложения специальный код, содержащий кусок SQL-запроса.

Такой вид атаки называется инъекция, в данном случае самый распространённый — SQL-инъекция. Это опаснейшая уязвимость, позволяющая злоумышленнику получить доступ к базе данных и возможность читать/изменять/удалять информацию, которая для него не предназначена. Например, изменить вместе с именем и фамилией баланс своего счета, посмотреть баланс чужого счета, или же, похитить конфиденциальные личные данные.

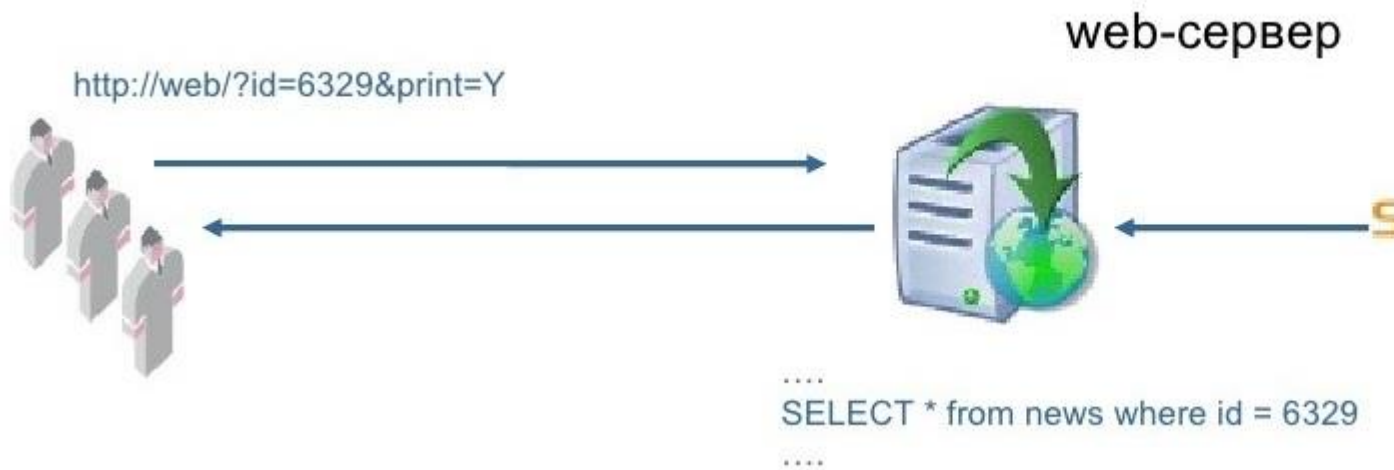
Эта уязвимость является следствием недостаточной проверки данных, поступающих от пользователя. Это позволяет злоумышленнику «подсунуть», например, в веб-формы, специально подготовленные запросы, которые «обманут» приложение и позволят прочесть или записать нелегитимные данные.

В целом эта разновидность атак имеет общее название «Ошибки валидации», к ней относятся далеко не только SQL-инъекции и мы будем упоминать этот вектор еще не раз.

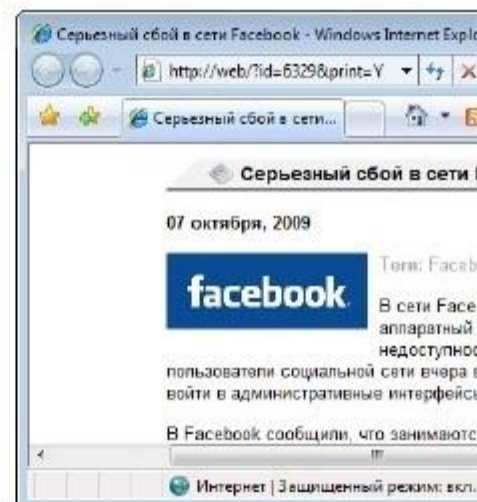
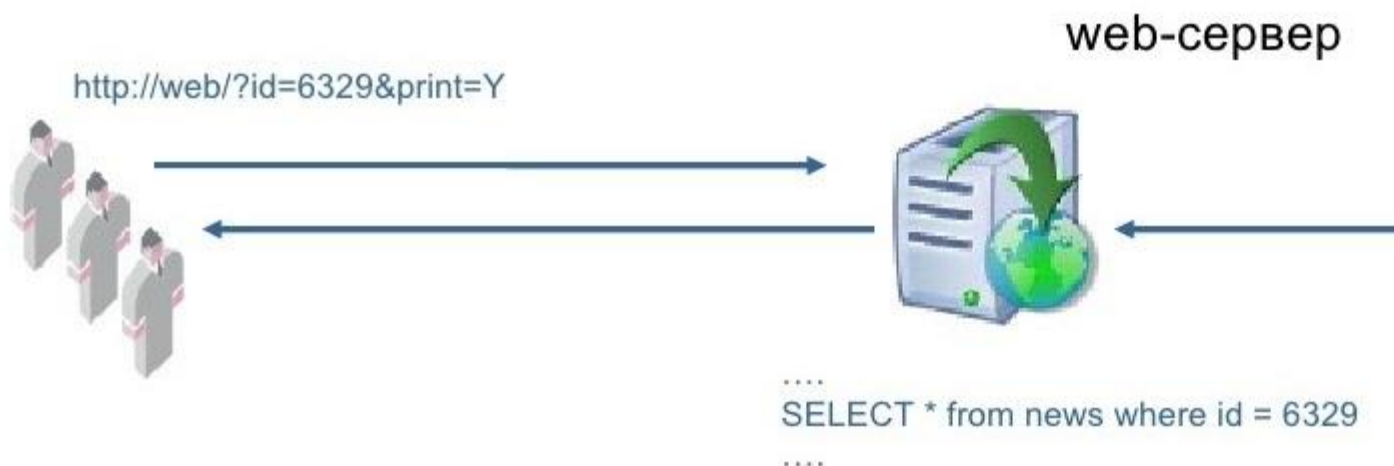
1. Пользователь переходит по ссылке, передавая определённые параметры



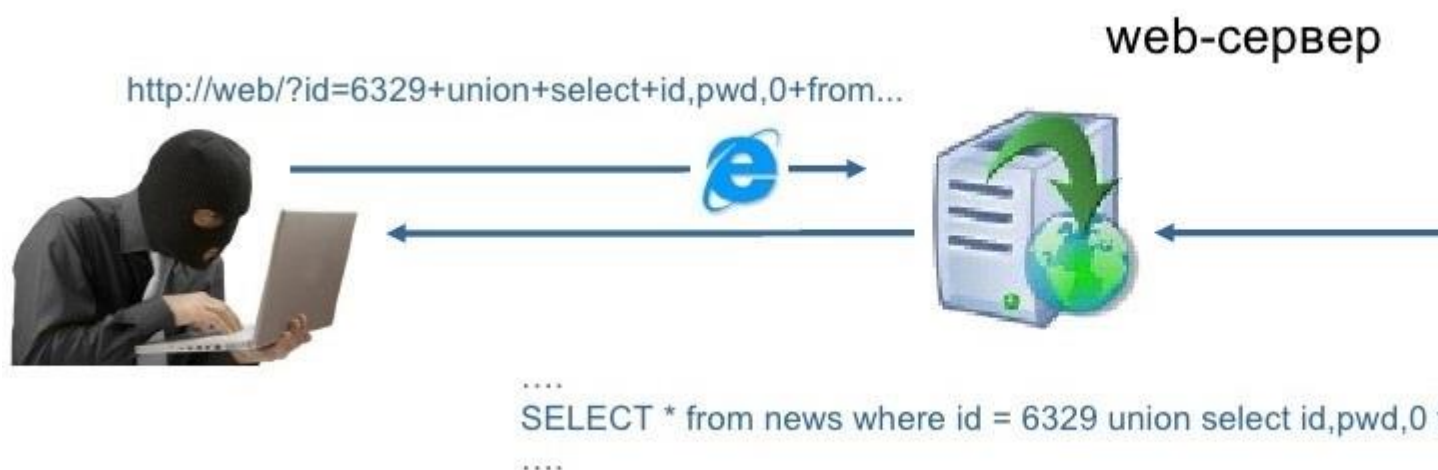
2. Сервер обрабатывает параметры, формирует SQL-запрос и отправляет его в базу данных



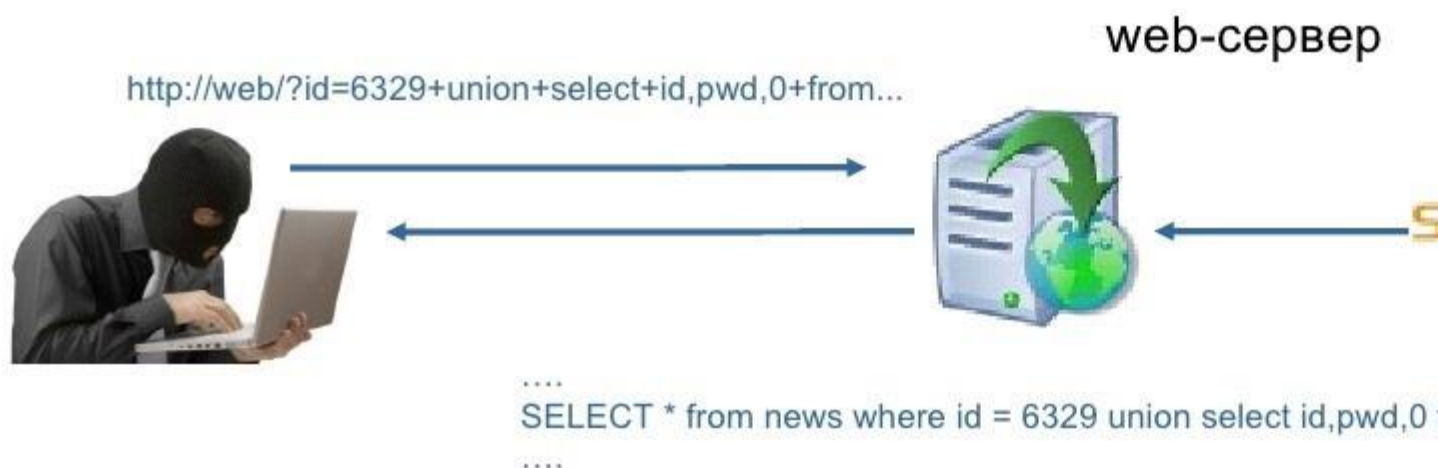
3. База данных возвращает результат и сервер использует эти данные в формировании страницы



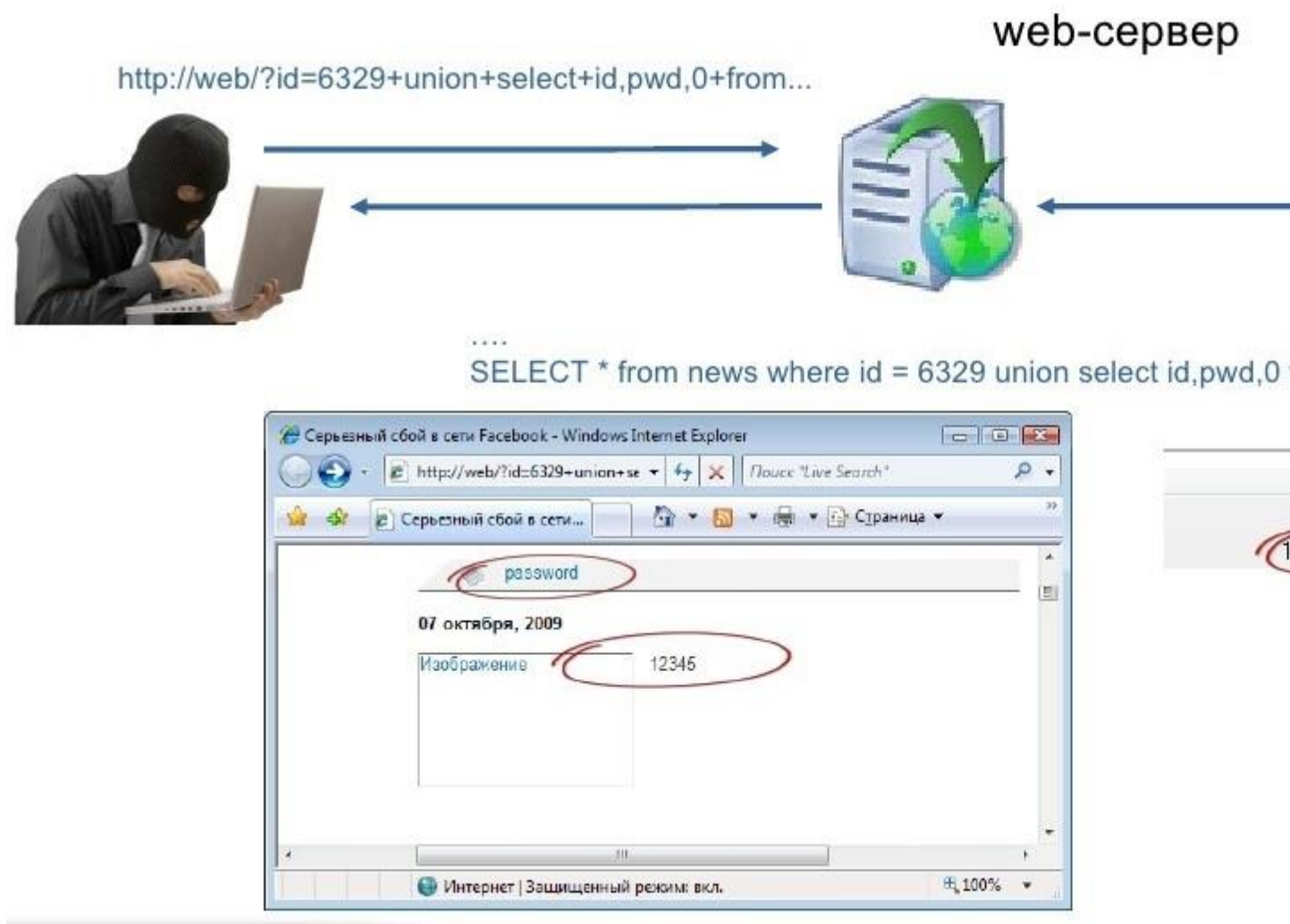
4. Что произойдёт, если в параметре передать строку, содержащую специальные символы, дополняющие SQL-запрос?



5. Сервер обработает эти данные, подставит их в запрос и, если запрос валидный, выполнит запрос



6. Далее данные обработаются и вернуться на страницу пользователя



Различают несколько видов SQL-инъекций:

1. Внедрение операторов SQL (SQL Injection)
2. Слепое внедрение операторов SQL (Blind SQL Injection)

Дополнительные материалы по атакам инъекции SQL

- Изложение технологии инъекции кода SQL [<https://www.dofactory.com/sql/injection>]
- Учебный сайт с уязвимостями SQL инъекций: [<https://redtiger.labs.overthewire.org>]
- Сайт с объяснением, как внедрять инъекции в учебный сайт: [<https://aayushmalla56.medium.com/sql-injection-redtigers-hackit-4f7b9399f1b0>]

1.3. Межсайтовый скриптинг (XSS)

Межсайтовое выполнение сценариев (*Cross site scripting* или XSS) – это возможность вставки HTML/JavaScript – кода в уязвимую страницу. Инъекция кода осуществляется через все доступные способы ввода информации: в полях пользовательского ввода, именах загружаемых файлов, параметрах HTTP запроса и т.п. Успешное завершение атаки может привести к использованию значений различных переменных, доступных в контексте сайта, записи информации, перехвату пользовательских сессий, данных и т.д. Cross-Site Scripting (XSS) – Это уязвимость на стороне клиента.

□

Условно Cross-Site Scripting (XSS) делят на:

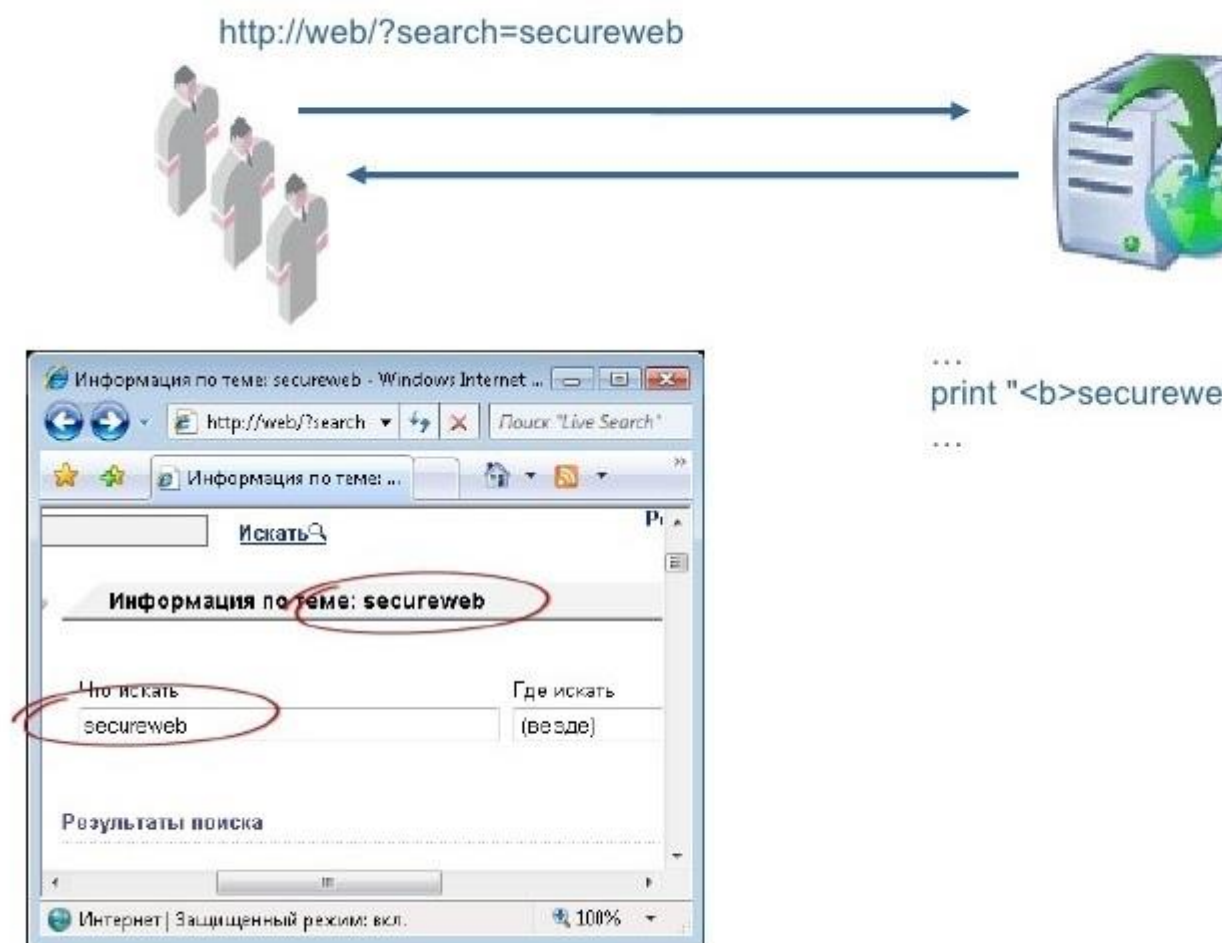
- Сохраненный вариант (Persistent / Stored)
- Отраженный вариант (non-persistent / reflected)

Как это работает:

1. Пользователь вводит на Web-сайте в строке поиска слово “secureweb”.

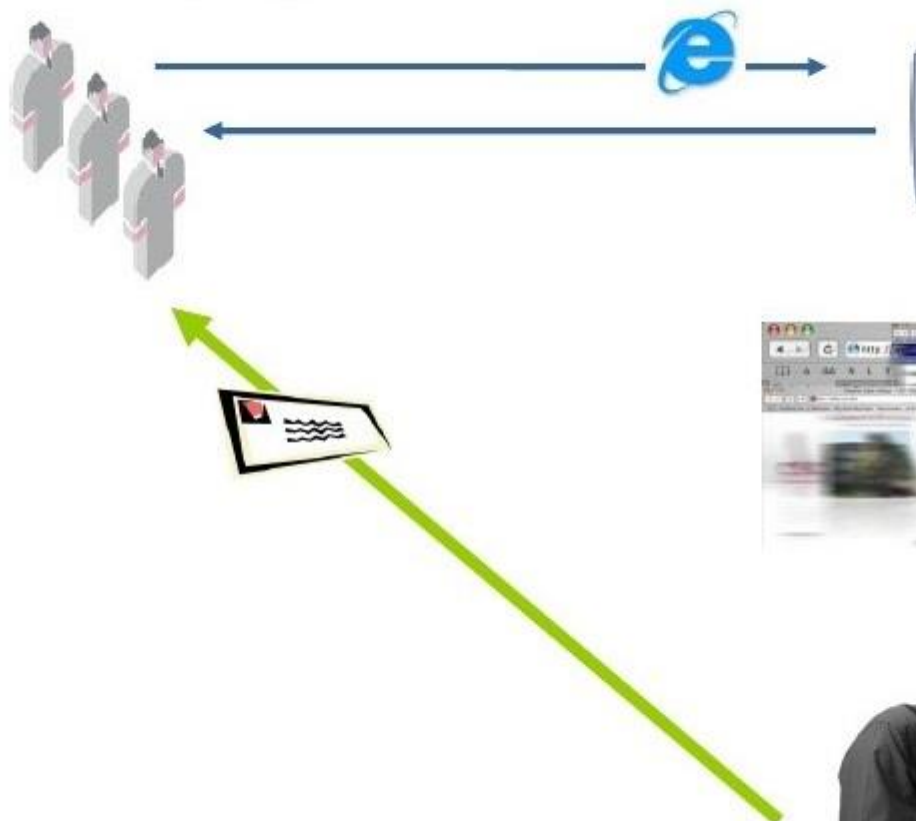


2. Web-сервер отображает введенную поисковую фразу на странице так, как она была введена в поле



3. Обнаружив уязвимость злоумышленник отправляет жертве ссылку с внедренным кодом

3. Переход по ссылке



2. Передача «заряженной» ссылки:

`http://web/?search=secureweb"><script>...</script>`

4. После перехода по ссылке в браузере пользователя выполняется внедренный код

3. Переход по ссылке



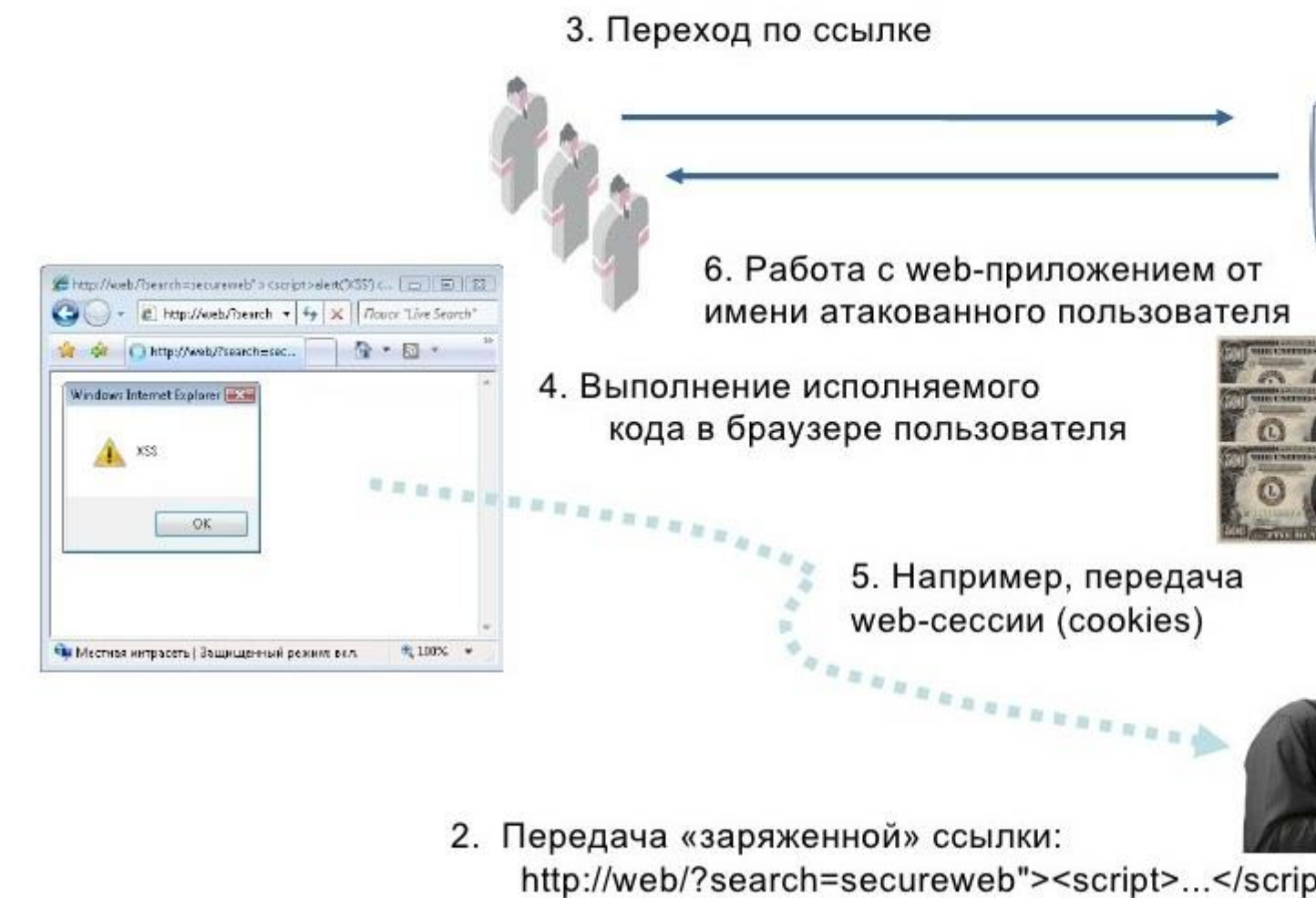
4. Выполнение исполняемого кода в браузере пользователя



2. Передача «заряженной» ссылки:

`http://web/?search=secureweb"><script>...</script>`

5. После выполнения кода злоумышленник может похитить авторизационные данные пользователя и выполнять действия от его имени на сайте



Дополнительные материалы по атакам XSS

- Основы XSS по-русски [<https://hackware.ru/?p=1174>]
- Основы XSS по-английски [<https://learn.snyk.io/lessons/xss/javascript/>]

1.4. Подбор паролей

Одним из основных механизмов защиты современных веб-приложений является механизм управления доступом. Обычно выделяют следующие этапы управления доступом:

- идентификация – установление идентификационных данных;
- аутентификация – подтвержденное установление идентификационных данных;
- авторизация – назначение прав идентификационным данным.

При входе в веб-приложение (sign in, log in) пользователь идентифицируется (сообщает свой идентификатор) и аутентифицируется (доказывает, что он именно тот пользователь, чей идентификатор был сообщен). Большинство веб-приложений используют аутентификацию по паролю. В веб-приложениях с высоким уровнем защищенности (например, в Интернет-банках) также применяются протоколы двухфакторной аутентификации. Очевидным недостатком аутентификации по паролю является возможность использования паролей с плохими

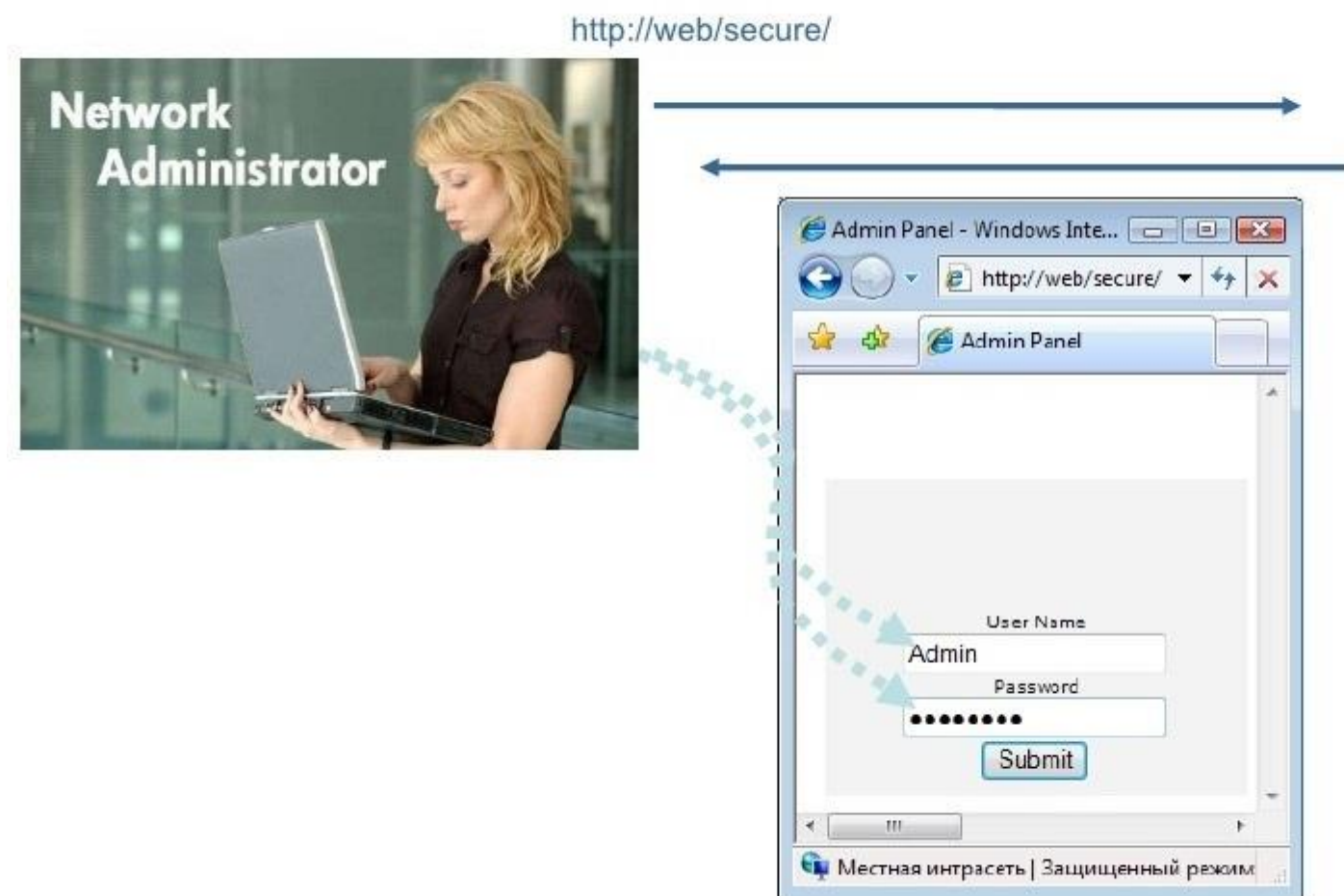
□

статистическими характеристиками. Хранение пароля или его передача по каналам связи в открытом или даже зашифрованном виде потенциально несет угрозу раскрытия пароля.

Очень большое количество пользователей использует очень простые и предсказуемые пароли для приложений. Очень часто пароль один от различных сервисов (почта, социальные сети, онлайн-банк).

Самой распространенной атакой является брутфорс или перебор по словарю. Как это работает:

1. Пользователь задаёт пароль и логин (или использует логин по умолчанию):



2. Злоумышленник с помощью специализированного софта последовательно подбирает пароли пользователя.

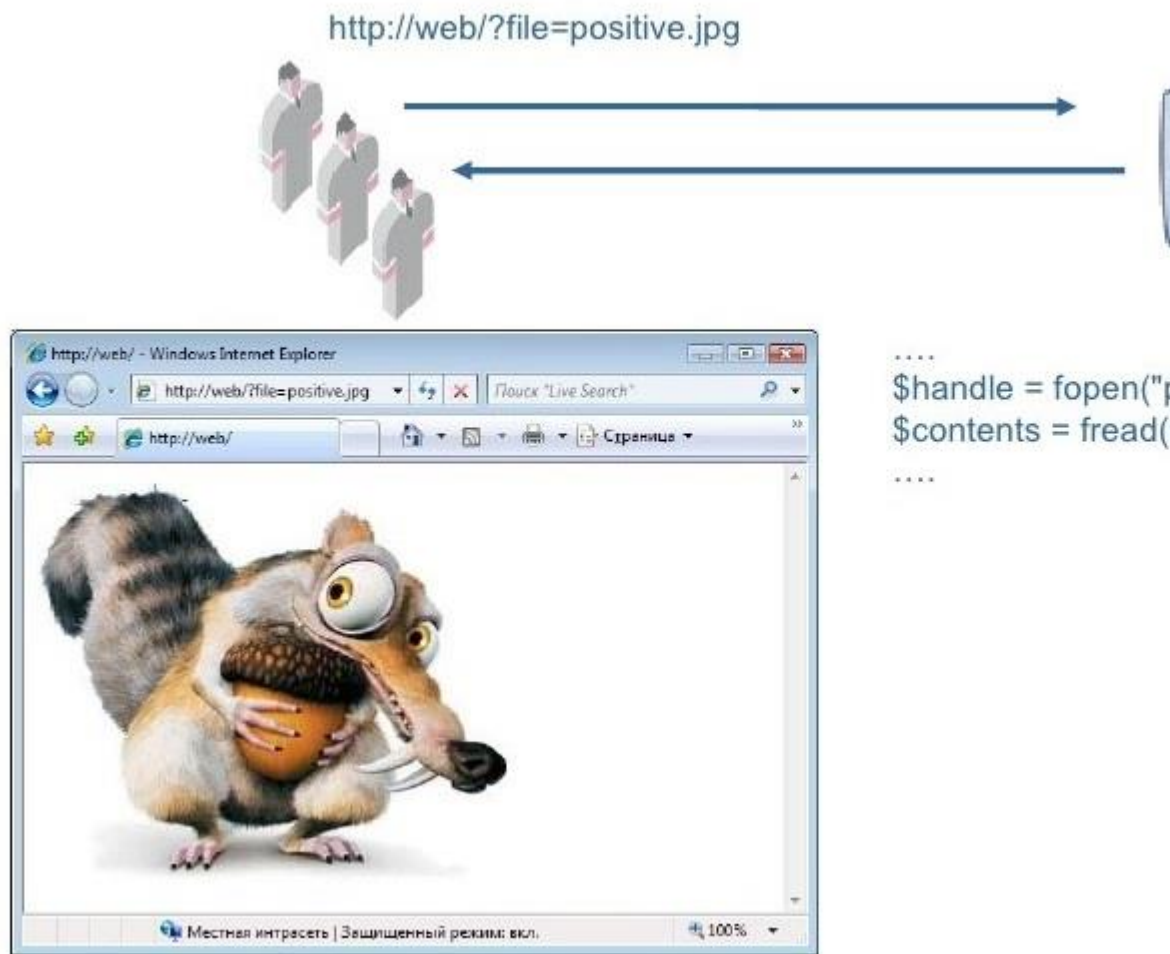


1.5. Обратный путь в директориях (Path Traversal)

Атака по обходному пути (также известная как *Path Traversal*) предназначена для доступа к файлам и каталогам, которые хранятся вне корневой папки сервера. Путем манипулирования переменными, которые ссылаются на файлы с чередованием «dot-dot-slash или (../)» и его вариантами или с использованием абсолютных путей к файлу, может быть получен доступ к произвольным файлам и каталогам, хранящимся в файловой системе, включая исходный код приложения или конфигурацию и критические системные файлы. Следует отметить, что доступ к файлам ограничен управлением операционным доступом системы (например, в случае заблокированных или используемых файлов в операционной системе Microsoft Windows) или файлах без права чтения в Linux.

Как это работает:

1. Сервер принимает запрос от пользователя, в котором содержится наименование файла, который необходимо открыть. Сервер обрабатывает параметр и возвращает пользователю файл с соответствующим названием.



2. Злоумышленник меняет наименование файла на путь к файлу, который он хочет прочесть, к примеру ../../../../../../../../../../etc/passwd. Сервер так же обрабатывает параметр и возвращает пользователю файл с соответствующим названием.

`http://web/?file=../../../../../../etc/passwd`



```
....
$handle = fopen("../../../../../../etc/passwd", "r");
$contents = fread($handle, 1024);
....
```

1.6. Прямой доступ к объектам

Данный вид уязвимости является также следствием недостаточной проверки пользовательских данных. Суть ее заключается в том, что при выводе каких-либо конфиденциальных данных, например личных сообщений или учетных карточек клиентов, для доступа к объекту используется идентификатор, который передается в открытом виде в адресной строке браузера или в параметрах запроса и не реализована проверка прав доступа к объектам. Например, есть страница, которая отображает личное сообщение и она имеет адрес вида:

`mysite.ru/read_message.jsp?id=123654`

Перебирая число после `id=` можно будет читать чужие личные сообщения. Эксплуатация данной уязвимости очень проста и не требует вообще никаких специальных навыков – достаточно лишь перебирать число в адресной строке браузера и наслаждаться результатом. Так же возможно перебирать имена файлов или другие идентификаторы. Как ни парадоксально, но этой "детской болезни", порой были подвержены достаточно крупные европейские платежные системы.

1.7. Некорректная настройка механизмов безопасности

После разработки приложения настает время правильно настроить общее окружение и сервера на которых будет эксплуатироваться данный софт. К сожалению, часто администраторы пренебрегают основными механизмами безопасности, что может привести к появлению

□

уязвимостей. Общие ошибки, относящиеся к данной категории можно выделить в следующие типы:

- доступность интерфейсов управления в т.ч. фактически неиспользуемые: сетевые порты, URI
- разглашение информации в сообщениях об ошибках (фреймворк, ЯП, СУБД, сервер)
- Проблемы на границе сфер ответственности администраторов и разработчиков (workingCopy)
- автодополнение и кэширование
- отсутствие флагов или заголовков безопасности (защита от client-side атак: Clickjacking, XSS)
- файлы, содержащие конф. информацию, доступны всем (*.php.1)

Несколько примеров того, как это работает:

- доступность интерфейсов управления

□

```
Host is up (0.0000050s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 6.0p1 Debian 4 (protocol 2.0)
3306/tcp   open  mysql    MySQL 5.5.33-0+wheezy1
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```



Welcome to phpMyAdmin

Language

English

Log in

Username: root

Password:

Go

- Разглашение информации в сообщениях об ошибках

□

Warning: ociexecute(): OCISstmtExecute: ORA-20000: Oracle Text-Fehler: DRG-11701: Thesaurus Oracle Database Enterprise Edition Release 11.1.0.7.0 - Production ist nicht vorhanden ORA-06512: in "CTXSYS.DRUE", Zeile 160 ORA-06512: in "CTXSYS.DRITHSX", Zeile 538 ORA-06512: in Zeile 1 in C:\app\as1\ohs\htdocs\php1.php on line 45

Warning: ocifetch(): OCIFetch: ORA-24374: define not done before fetch or execute and fetch in C:\app\as1\ohs\htdocs\php1.php on line 45

java.lang.NullPointerException

```
stacktrace.clj:15 ring.middleware.stacktrace/wrap-stacktrace-log[fn]
stacktrace.clj:79 ring.middleware.stacktrace/wrap-stacktrace-web[fn]
reload.clj:18 ring.middleware.reload/wrap-reload[fn]
jetty.clj:16 ring.adapter.jetty/proxy-handler[fn]
(Unknown Source) ring.adapter.jetty.proxy$org.mortbay.jetty.handler.AbstractHandler$0.handle
HandlerWrapper.java:152 org.mortbay.jetty.handler.HandlerWrapper.handle
Server.java:326 org.mortbay.jetty.Server.handle
HttpConnection.java:542 org.mortbay.jetty.HttpConnection.handleRequest
HttpConnection.java:926 org.mortbay.jetty.HttpConnection$RequestHandler.headerComplete
HttpParser.java:549 org.mortbay.jetty.HttpParser.parseNext
HttpParser.java:212 org.mortbay.jetty.HttpParser.parseAvailable
HttpConnection.java:404 org.mortbay.jetty.HttpConnection.handle
SocketConnector.java:228 org.mortbay.jetty.bio.SocketConnector$Connection.run
QueuedThreadPool.java:582 org.mortbay.thread.QueuedThreadPool$PoolThread.run
```

- Проблемы на границе сфер ответственности администраторов и разработчиков (workingCopy)

Index of /jp/.svn

- [Parent Directory](#)
- [all-wcprops](#)
- [all-wcprops1111](#)
- [entries](#)
- [format](#)
- [prop-base/](#)
- [props/](#)
- [text-base/](#)

Index of /.git

- [Parent Directory](#)
- [HEAD](#)
- [config](#)
- [description](#)
- [hooks/](#)
- [index](#)
- [info/](#)
- [logs/](#)
- [objects/](#)
- [packed-refs](#)
- [refs/](#)
- [shallow](#)

1.8. Использование компонентов с известными уязвимостями

Ни одно современное веб-приложение не работает без подключения сторонних компонентов (пришлось бы тратить слишком много времени на изобретение велосипеда). Основная проблема данного подхода в том, что приложение использует чужой код, но не следит за его актуальностью и состоянием. А в этом коде могут содержаться критические уязвимости или незадекларированные возможности. Очень хорошим примером является пример с библиотекой SSL/TLS, в которой за последние несколько лет нашли огромное количество уязвимостей:

- FREAK (экспортные ключи)
- POODLE (downgrade и CBC)
- HEARTBLEED (некорректная проверка длины)
- BEAST (предсказуемый IV и CBC)
- CRIME / BREACH (компрессия)

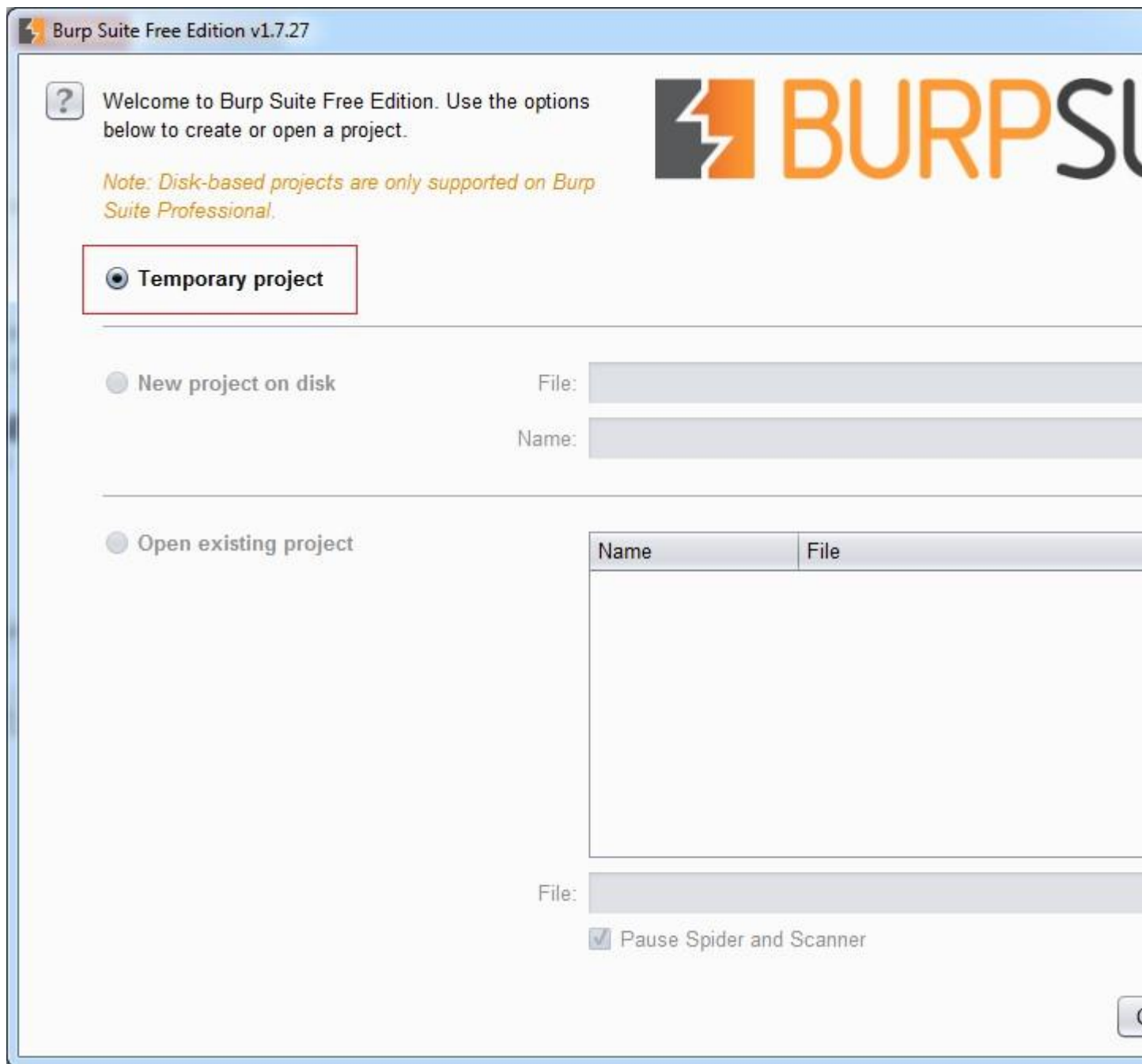
2. Практическая часть

□

2.1. Настройка Burp Suite

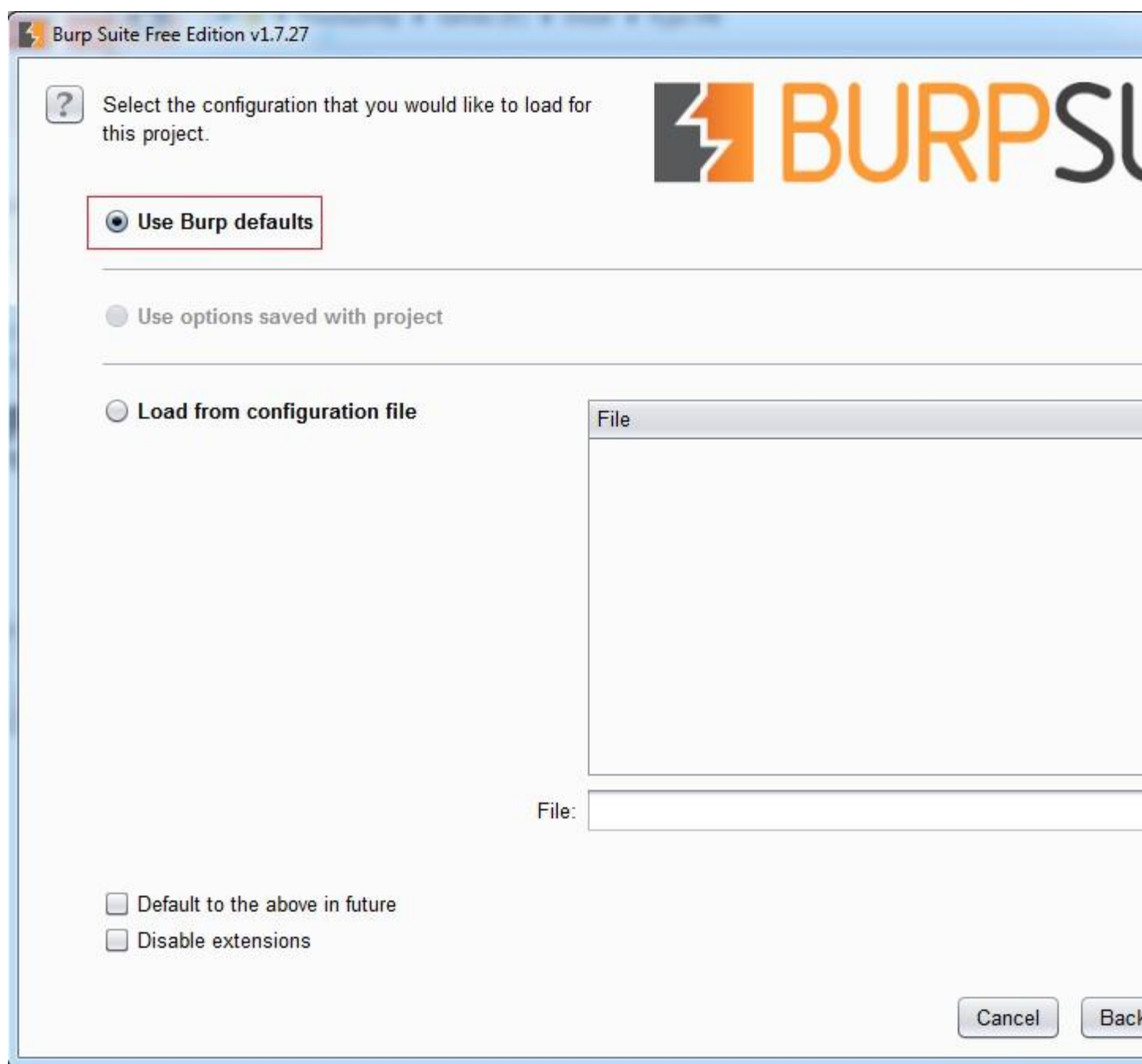
Скачайте и установите [Burp Suite Community](#)

Создайте временный проект



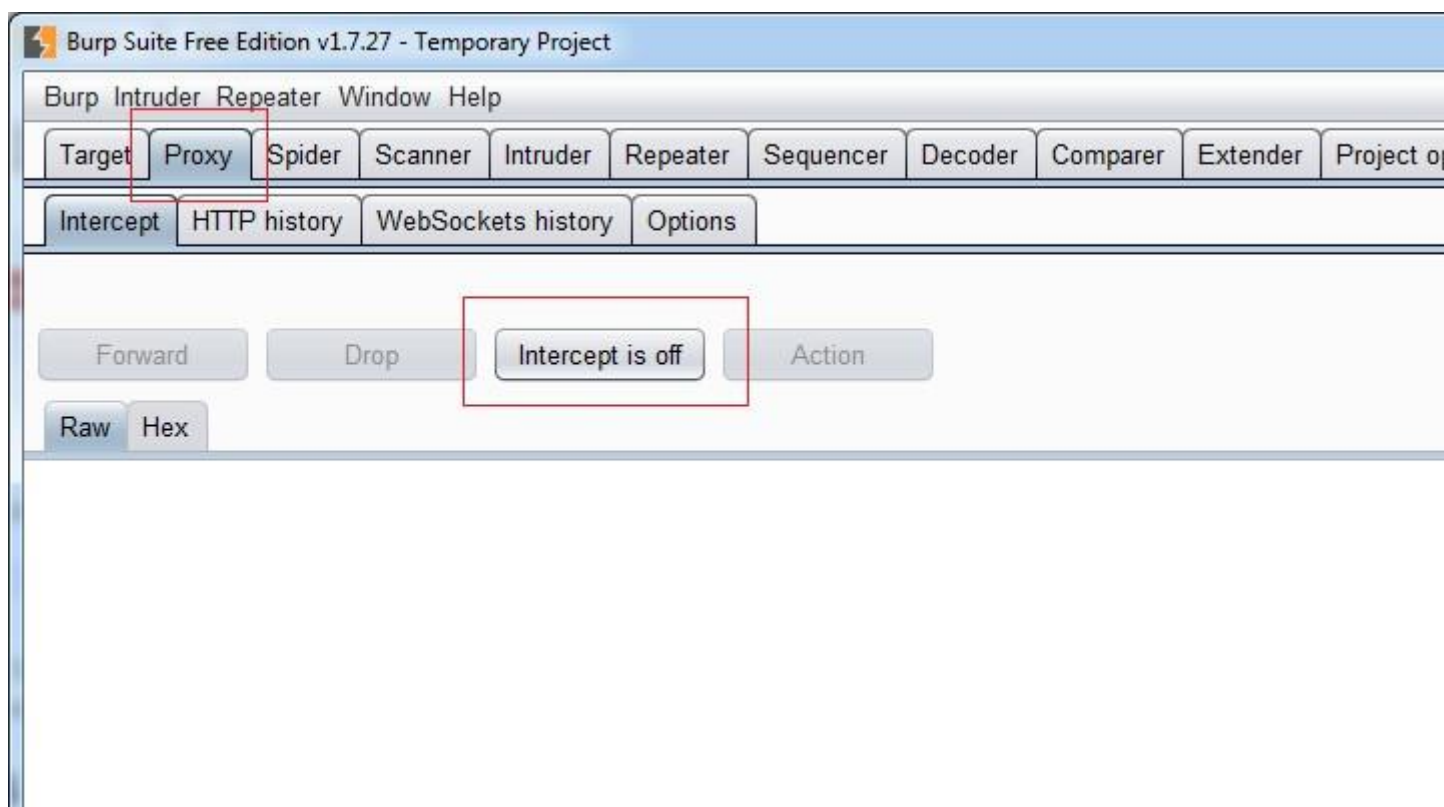
Выбираем настройки по умолчанию

□

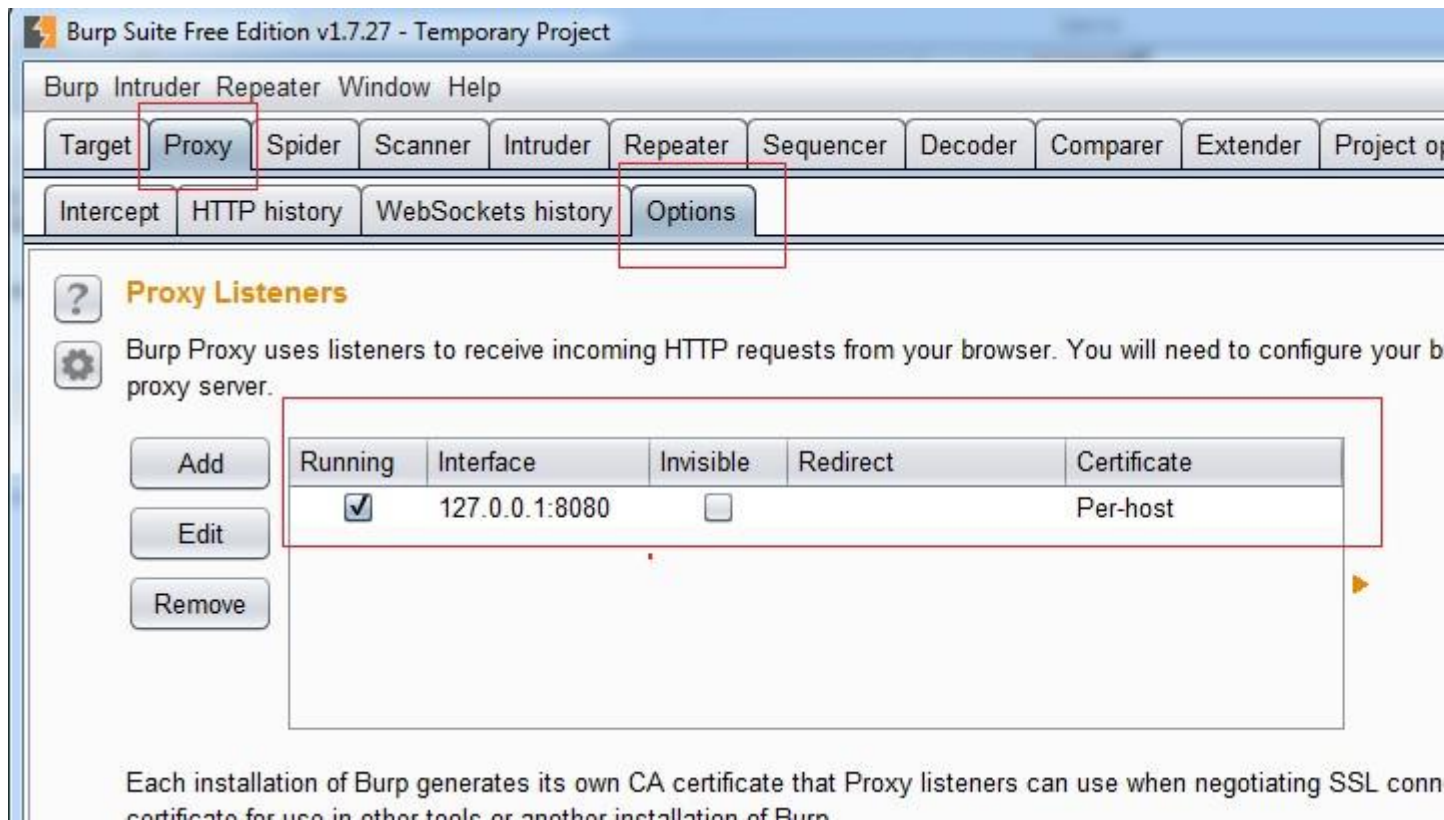


После запуска программы переходим на вкладку “Proxy”, отключаем прерывание запросов:

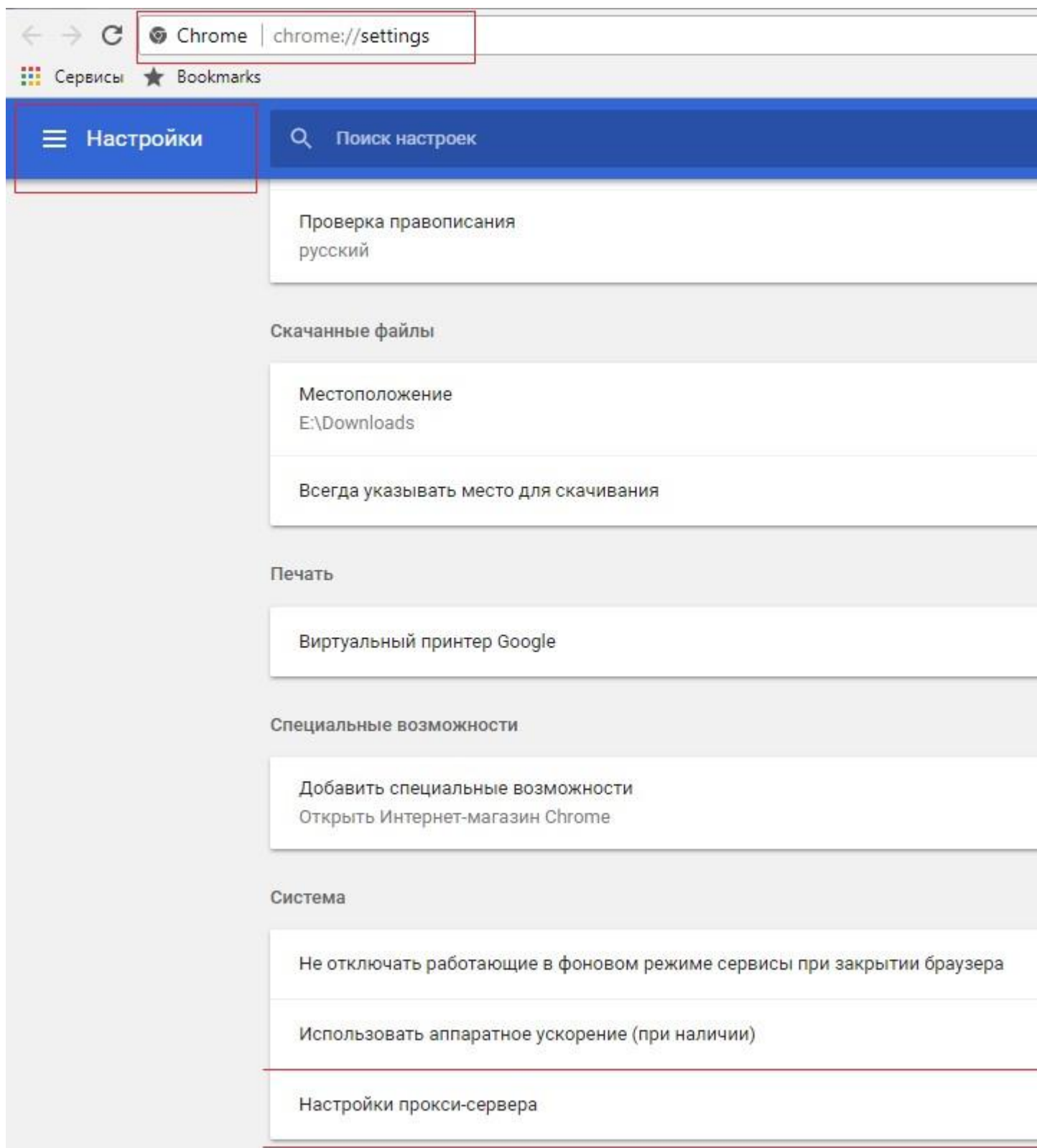
□



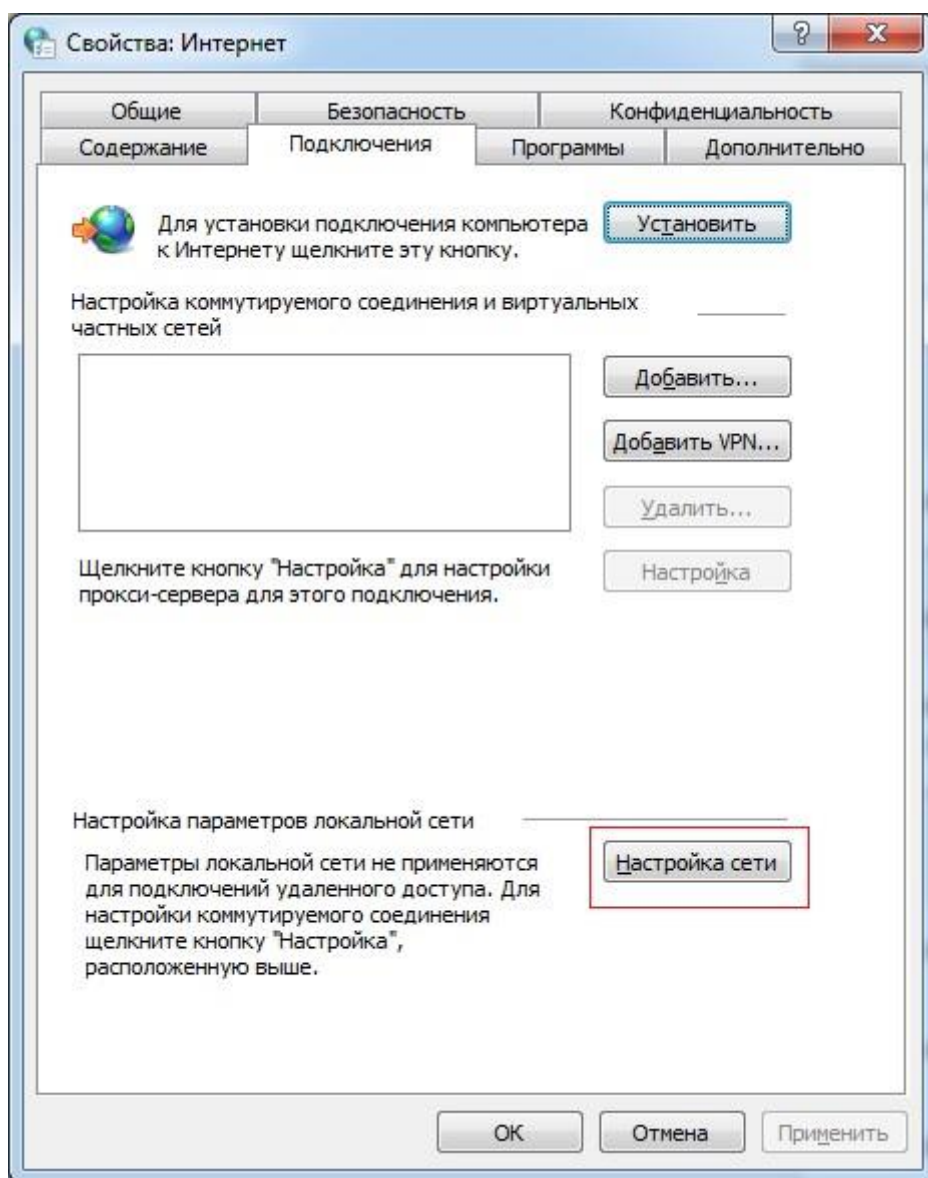
Переходим на вкладку «Options». Настройка прокси-сервера должна выглядеть следующим образом:



Открываем браузер Chrome и заходим в «настройки» -> «Дополнительно» -> «Настройки прокси-сервера»

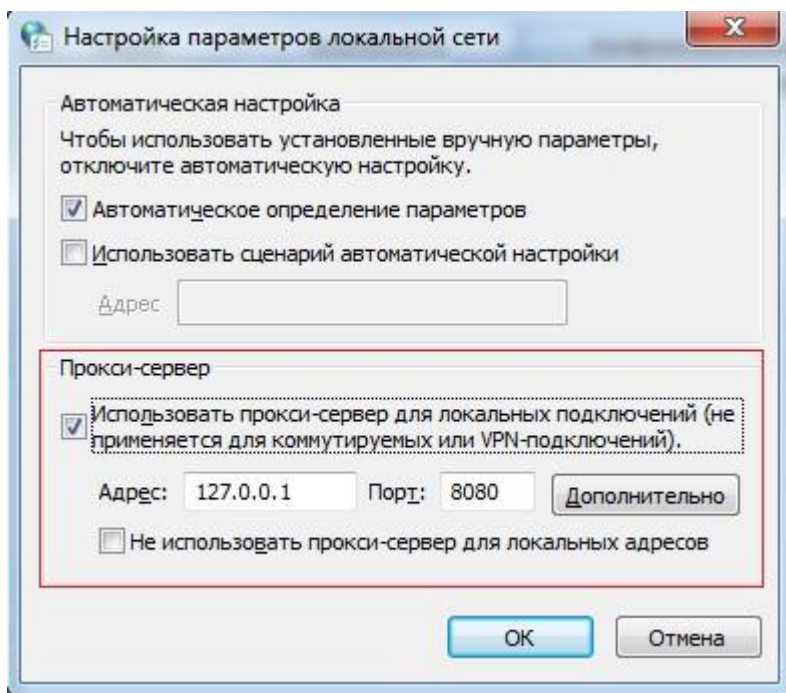


После нажатия, попадаем в меню Подключений. Выбираем «Настройка сети»



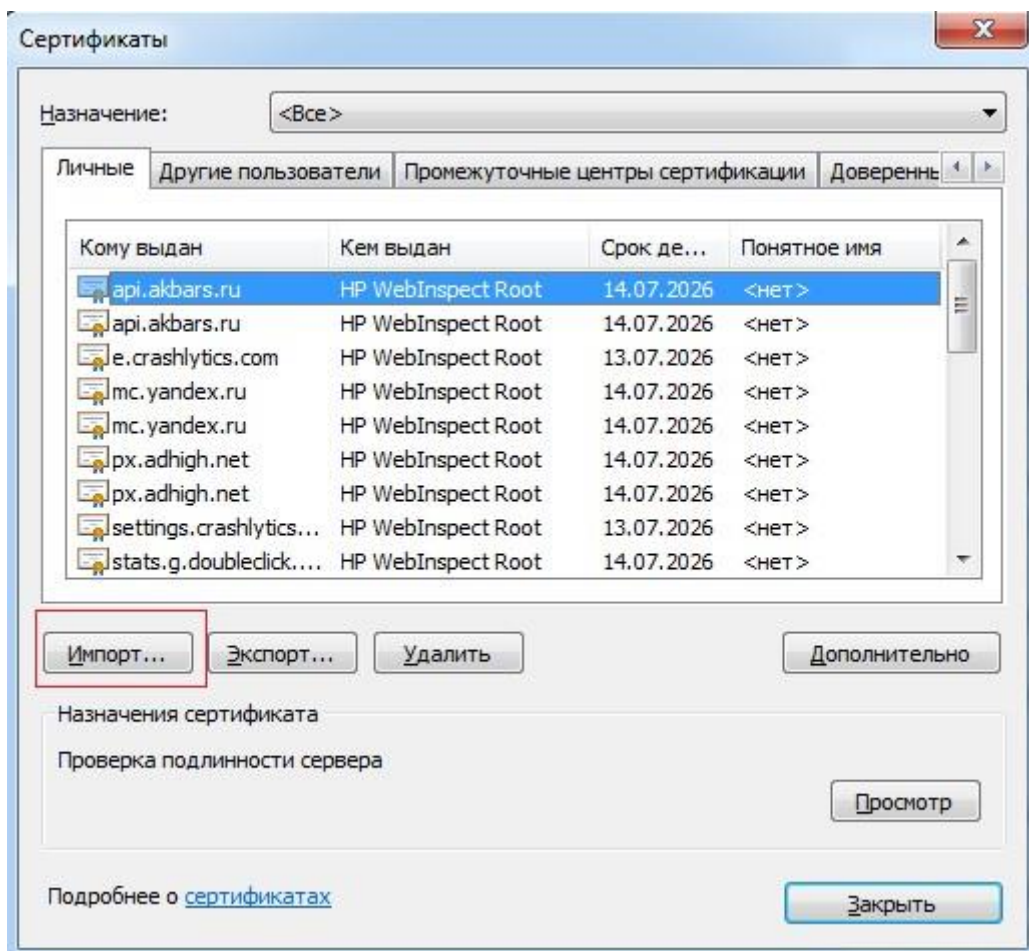
Настраиваем прокси-сервер в соответствии с тем, что было указано в настройках Burp Suite.

□



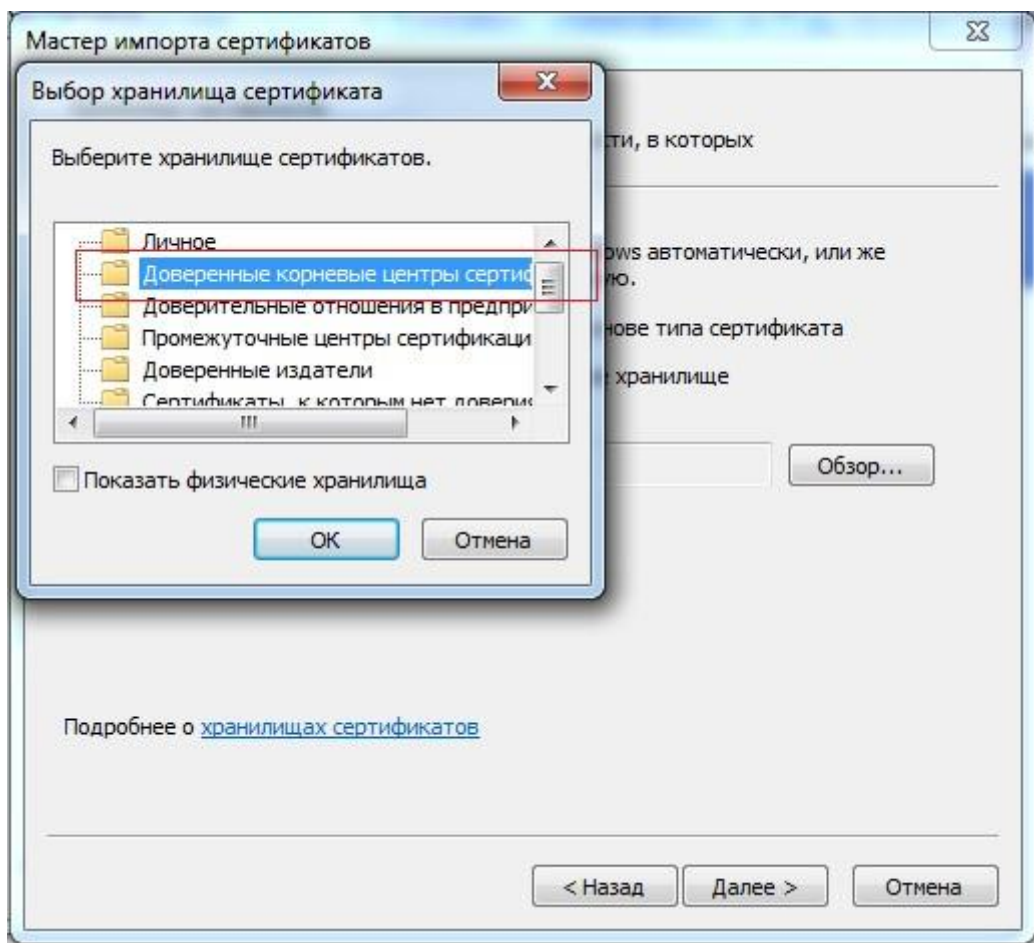
После настройки переходим на сайт <http://burpsuite/> и скачиваем сертификат, нажав на кнопку CA Certificate.

После этого заходим в «настройки браузера Chrome» → «Дополнительно» → «Настроить сертификаты».



□

В появившемся окне нажимаем импорт и продвигаемся по меню, выбрав скачанный ранее файл сертификата. Его необходимо добавить в доверенные центры сертификации.



Настройка завершена. Для того, чтобы удостовериться, что всё работает, введите в адресной строке <http://yandex.ru>

После отображения страницы перейдите в Burp Suite на вкладку «Proxy» -> «HTTP History». В ней вы увидите запросы, которые отправляет ваш браузер и ответы, которые получает от сервера.

Burp Suite Free Edition v1.7.27 - Temporary Project

Burp Intruder Repeater Window Help

Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Project of

Intercept HTTP history WebSockets history Options

Filter: Hiding CSS, image and general binary content

#	Host	Method	URL	Params	Edited	Status	Length
2562	https://www.google.ru	GET	/_chrome/newtab-serviceworker.js	☐	☐	200	4084
2563	https://www.google.ru	GET	/complete/search?client=chrom...	☒	☐	200	1346
2564	https://www.google.ru	GET	/complete/search?client=chrom...	☒	☐	200	1176
2565	https://www.google.ru	GET	/complete/search?client=chrom...	☒	☐	200	1197
2566	http://ya.ru	GET	/	☐	☐	302	425
2567	https://ya.ru	GET	/	☐	☐	200	23406
2568	https://yandex.ru	GET	/data/mail.js?yaru=y&cb=userInf...	☒	☐	200	952
2571	https://mc.yandex.ru	POST	/watch/723233?wmode=7&page-...	☒	☐	200	627
2574	https://s814.thetrafficstat....	POST	/related	☒	☐	204	109
2575	https://s1814.thetrafficstat...	POST	/service2	☒	☐	204	109

Request Response

Raw Params Headers Hex

```
GET / HTTP/1.1
Host: ya.ru
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Safari/537.36
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;
Accept-Encoding: gzip, deflate
Accept-Language: ru-RU,ru;q=0.8,en-US;q=0.6,en;q=0.4
Cookie: yandexuid=9318769161507637074; yp=1510229075.ygu.1; mda=0; yandex_gid=213; _ym_u
zm=m-white_yaru.css%3Awww_I9mAdLendvFe8dG8XF4eZ0BEFY%3A1; _ym_isad=2
Connection: close
```

? < + > Type a search term

*Открыть последний ответ и сделать скриншот

2.2. Внедрение кода (SQL Injection)

□

*

2.3. Подбор пароля (Brute-force attack)

Зайти на сайт `http://10.5.143.101:PORT/#/login`

Проверить, что отображается в Burp Suite. Войти с учётными данными в приложение (admin@juice-sh.op : admin123) Выйти из приложения.

Попытаться войти в приложение, используя неправильный пароль.

Изучить взаимодействие с сервером, определить критерий успешного или не успешного логина в приложение.

Выбрать запрос, отвечающий за логин пользователя, нажать правой кнопкой, выбрать пункт «Send to Intruder».

Перейти на вкладку «Intruder» -> «Position»

Target

Positions

Payloads

Options



Payload Positions

Configure the positions where payloads will be inserted

Attack type: **Sniper**

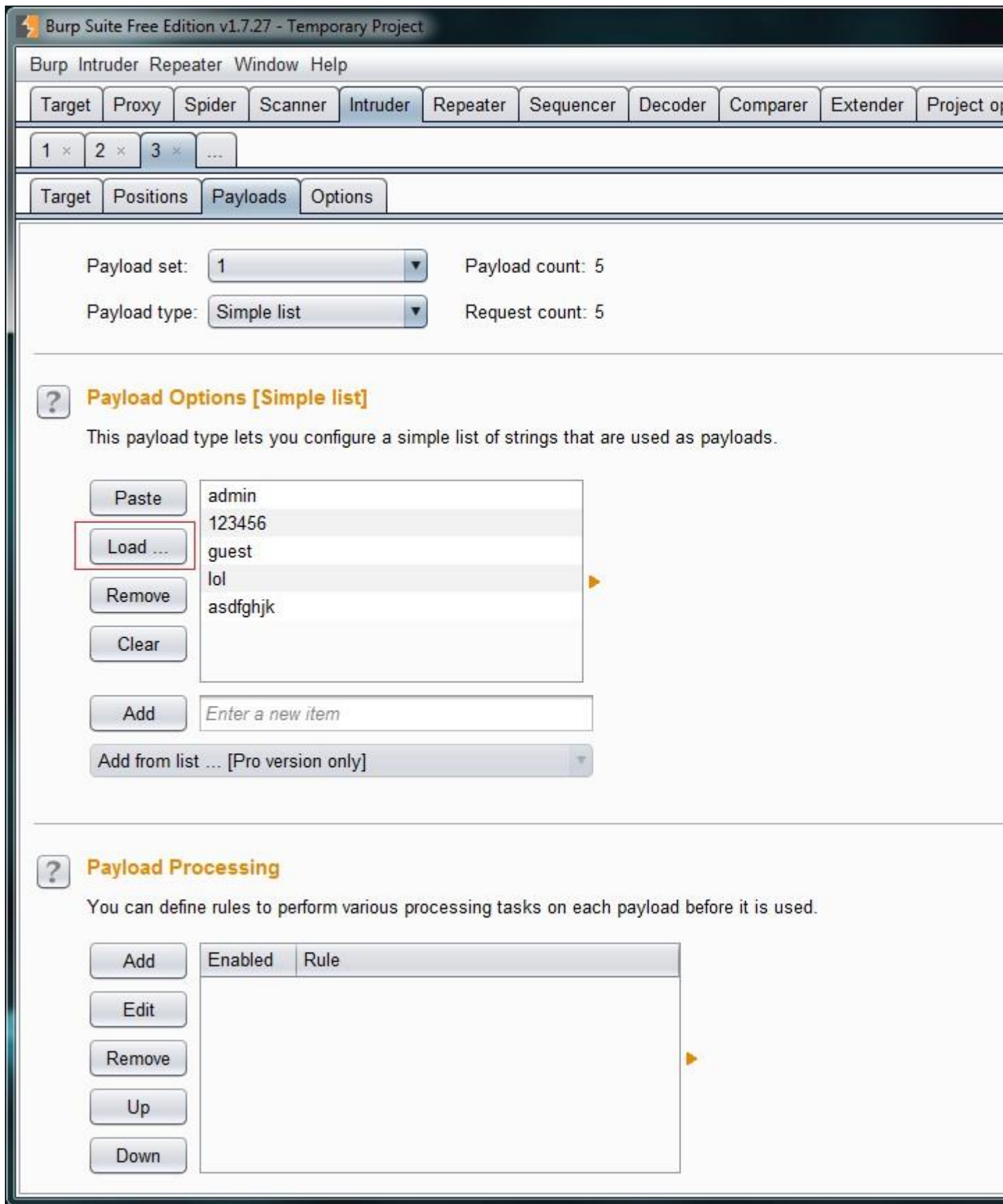
```
POST /rest/user/login HTTP/1.1
Host: juice-shop.herokuapp.com
Connection: close
Content-Length: 51
Accept: application/json, text/plain, */*
Origin: https://juice-shop.herokuapp.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac
Content-Type: application/json
Referer: https://juice-shop.herokuapp.com/
Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,e
Cookie: language=en; io=uOv34K1UsNcxwtG4AAD5
token=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1b290b3RtaW4iOnRydWUsImxhc3Rmb2dpbklwIjoiaWMC4wLjA6MTUuMzIzICswMDowMCJ9LCJpYXQiOiE1NTc3MzUwODUsYgdxYDh91NxBmA0AJdRaS7XHhicJK60

{"email":"admin@juice-sh.op","password":"$adr
```

□

В окне выделить символом § с двух сторон только форму password. Создать на компьютере файл, содержимое которого является паролями, которые будут перебираться (каждое значение с новой строки). В файле должен быть валидный пароль.

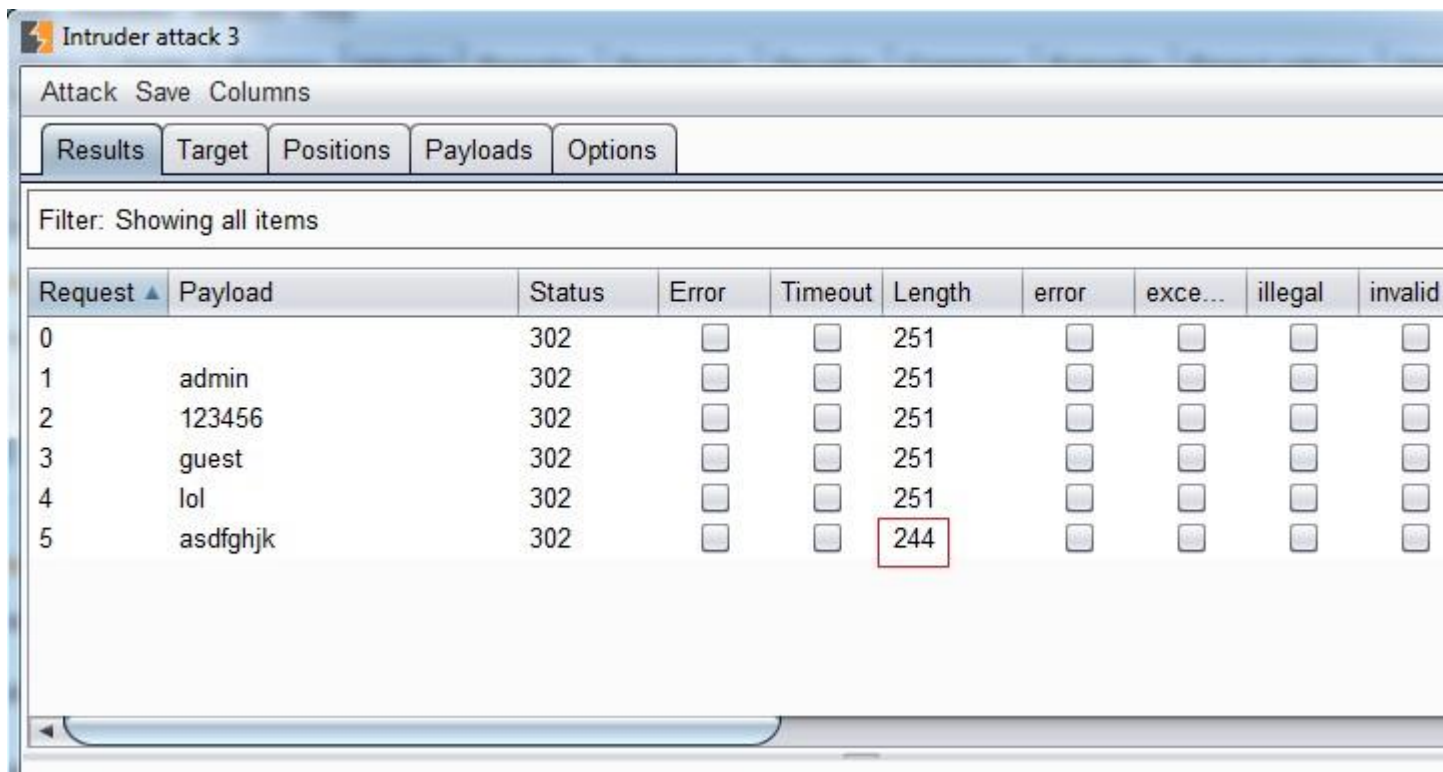
Перейти на вкладку «Payloads», загрузить файл через кнопку «Load...»



Нажать на кнопку «Start Attack», дождаться окончания. После в приведенной таблице проанализировать ответы сервера и понять, какой пароль из представленных действительно подошёл от нашей учётной записи.

□

Это можно сделать, посмотрев на длину ответа от сервера, для валидного пароля она будет отличаться.



Request	Payload	Status	Error	Timeout	Length	error	exce...	illegal	invalid
0		302	☐	☐	251	☐	☐	☐	☐
1	admin	302	☐	☐	251	☐	☐	☐	☐
2	123456	302	☐	☐	251	☐	☐	☐	☐
3	guest	302	☐	☐	251	☐	☐	☐	☐
4	lol	302	☐	☐	251	☐	☐	☐	☐
5	asdfghjk	302	☐	☐	244	☐	☐	☐	☐

***Скриншот**

2.4. Межсайтовый скриптинг (XSS)

Перейти на <https://xss-game.appspot.com/level1>

В появившемся окне в строке поиска ввести строку

```
<script>alert('1')</script>
```

Посмотреть на результат. В контексте браузера выполнен код, который мы передали в качестве поискового запроса.

***Скриншот**

Переходим на уровень 2.

Здесь представлен форум, на котором можно писать сообщения. При этом тэг `<script>` фильтруется и использовать его нельзя. Попробуем обойти другим способом и внедрим следующий код в страницу:

```

```

***Скриншот**

Таким образом мы попросили браузер обратиться за картинкой, которой не существует и, в случае ошибки, выполнить код указанный далее.

□

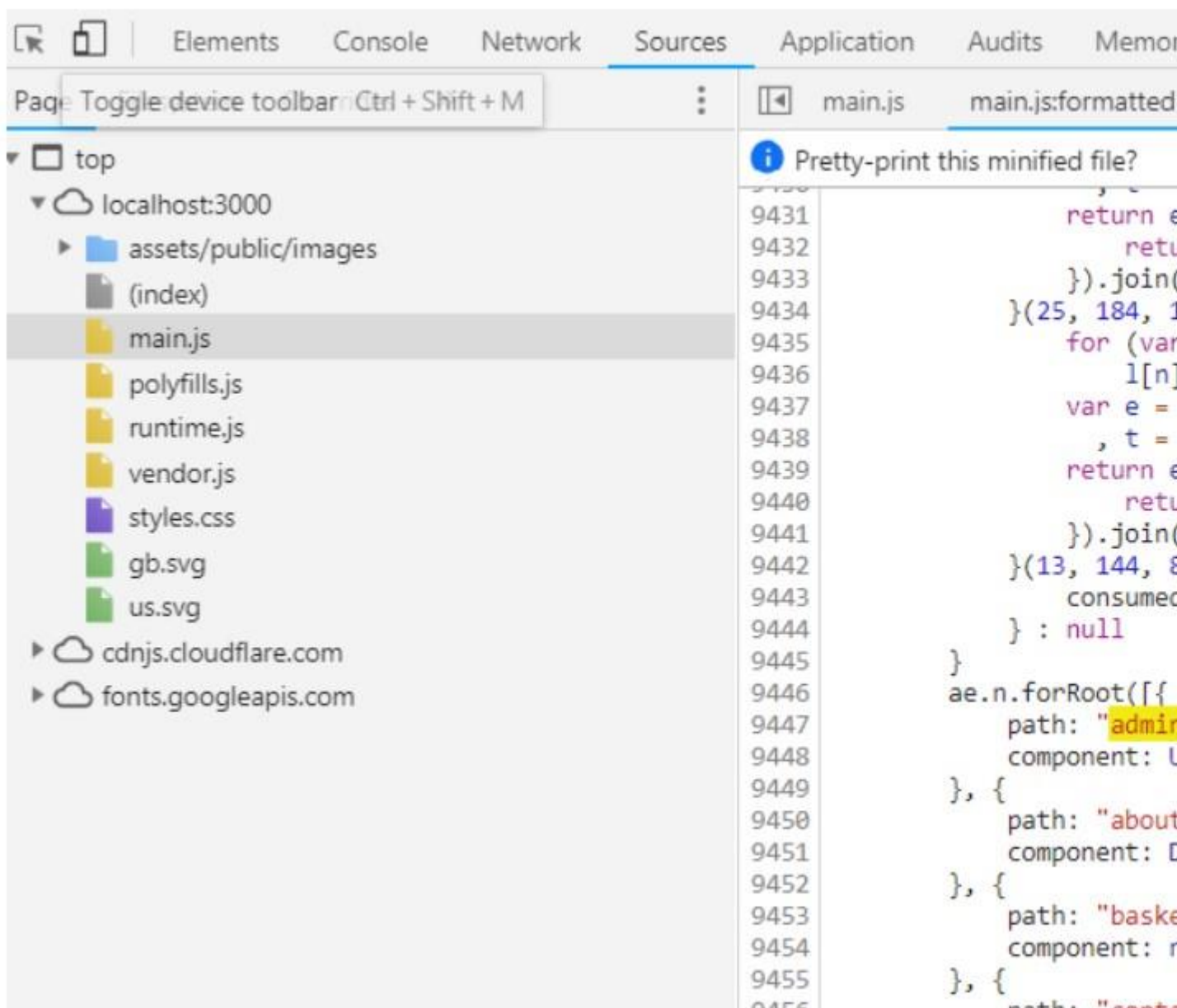
2.5. Прямой доступ к объектам

Зайдите на сайт `http://10.5.143.101:PORT/`

Попробуем найти панель администратора, которой нет в интерфейсе (нет кнопки, на которую можно нажать и перейти в админ-панель).

Для этого в браузере нажмем F12 и перезагрузим страницу. Теперь, необходимо найти файл `main.js` и провести в нем поиск по слову “admin”.

Увидим, что он встречается в строке для роутинга `path: "administration"`



Таким образом, обратившись напрямую к `http://10.5.143.101:PORT/#/administration` попадем в панель администратора

*Скриншот

□

3. Задание

На сайте `http://10.5.143.101:PORT/#/` найти и сделать скриншот:

- Cross-Site Scripting (XSS)
- SQL Injection
- Панель scoreboard

*дополнительно можно попробовать получить таблицу пользователей

Каждый скриншот должен сопровождаться подробным описанием вводимых команд/настроек и результатов их выполнения

Лабораторная работа №10

Тема: Подготовка презентации на заданную тему.

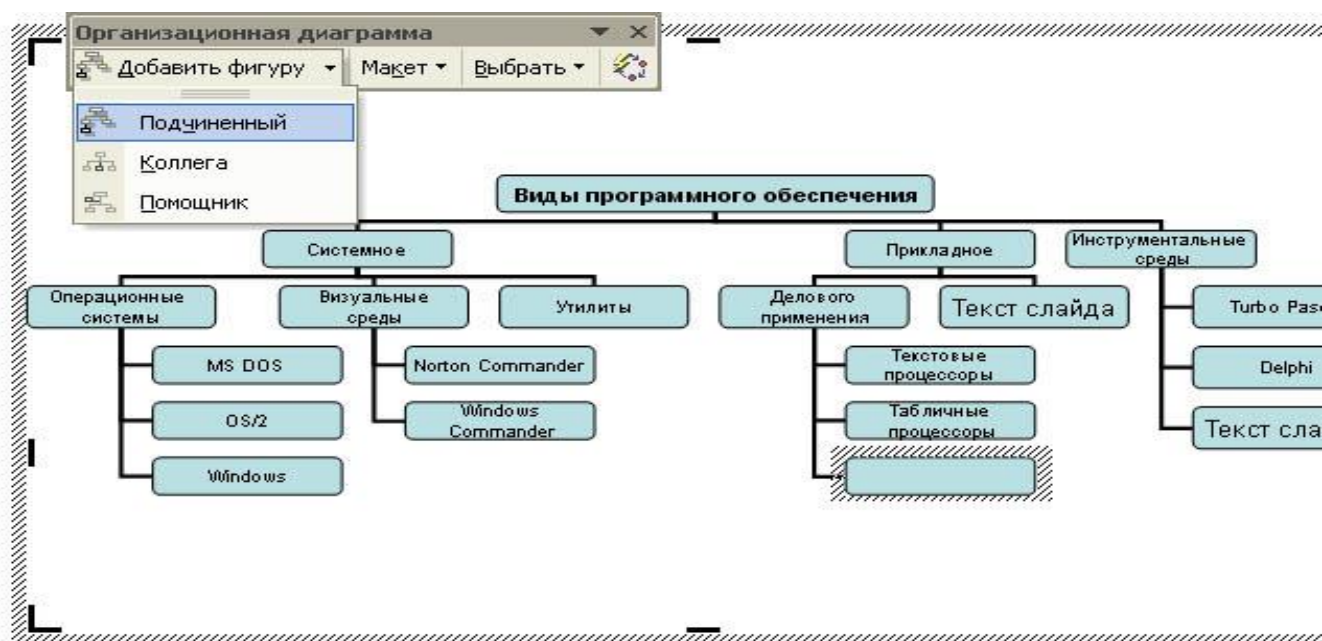
Цель: Создание презентации. Использование различных возможностей компьютерной презентации с помощью программы Power Point.

Задание 1. Создать презентацию на тему Виды компьютерных программ. Для этого выполните следующую последовательность действий.

1. Для создания презентации выберите команду **Создать** в меню **Файл**, а затем в панели задачи *Создать презентацию* щелкните ссылку *Новая презентация*. В окне *Создание слайда* выберите автомакет *Титульный слайд* и щелкните кнопку ОК.
2. Щелкнув мышью в области заголовка слайда, введите заголовок презентации *Виды компьютерных программ*. В подзаголовок слайда введите текст Программным обеспечением называют организованную совокупность программ постоянного употребления, ориентирующую ЭВМ на тот или иной класс применений. Используя инструменты панели *Форматирование*, установите нужные параметры текстам заголовка и подзаголовка.
3. Для вставки нового слайда с организационной диаграммой выберите в панели задач *Разметка слайда* макет **Объект**. Щелкнув на макете **Объект** стрелку справа, выберите в контекстном меню команду **Добавить новый слайд**

Щелкнув на новом слайде в панели *Рисование* Добавить организационную диаграмму.

Введите заголовок диаграммы *Типы программного обеспечения*, затем добавьте нужные фигуры и введите элементы диаграммы, как показано на рис. 3.



Для определения параметров текста можно использовать команды меню **Текст**. Для определения параметров элементов диаграммы применяйте команды контекстного меню **Формат Автофигуры** или инструменты из панели *Рисование*.

4. Для вставки нового слайда выберите в панели задачи *Разметка слайда* макет *Заголовок, текст и графика* и, щелкнув стрелку справа, выберите команду **Добавить новый слайд**. В область заголовка введите текст Прикладные программы. В область текстовой рамки слайда введите текст о назначении прикладных программ, например, Прикладное программное обеспечение -программы для решения класса задач в определенной области применения систем обработки данных. Они непосредственно обеспечивают выполнение необходимых пользователям работ.

В область графики вставьте картинку, для чего дважды щелкните кнопку *Вставка картинки* на слайде, а затем в раскрывшемся окне *Microsoft* выберите нужный рисунок. Можно вставить рисунок из файла, для чего, указав область рисунка на слайде, выберите в меню **Вставка** команду **Рисунок**

5. Действуя аналогично, добавьте слайды и введите текстовую и графическую информацию об остальных типах программного обеспечения.

Задание 2. Оформите презентацию и настройте анимацию объектов. Для этого выполните следующую последовательность действий.

1. Оформите презентацию с использованием готовых шаблонов оформления, для чего откройте в области задач панель *Дизайн слайда*. Выбирая в поле *Применить шаблон оформления* выберите подходящий шаблон.
2. Для настройки анимации слайдов откройте в области задач панель задачи *Настройка анимации*. Выбрав слайд и выбирая элемент слайда, настройте эффекты анимации. Для этого щелкните кнопку **Добавить эффект** и выполните одно или несколько следующих действий. Если требуется добавить определенный визуальный эффект в текст или объект, находящиеся на самом слайде, укажите значок *Выделение*, а затем выберите нужный эффект. Если требуется добавить определенный визуальный эффект в текст или объект, который вызывает удаление текста или объекта со слайда в заданный момент, выберите значок *Выход*, а затем - нужный эффект

Для просмотра заданного эффекта анимации щелкните кнопку **Просмотр**. Для изменения порядка появления анимации или ряда анимированных фрагментов, выбрав параметр в списке настроек анимации, перетащите его в другое место списка.

Задание 3.

Создать презентацию по одной из пройденных ранее тем, записанных в тетрадах, прим

Лабораторная работа № 11

Тема : Доработка презентации: вставка заданных объектов.

Создать презентацию из 10-15 слайдов, отражающие наиболее значимые открытия российских ученых в любых фундаментальных или прикладных научных исследованиях в области техники, педагогики, лингвистики, психологии и т.д.

На слайдах должны быть отображены: портрет, фамилия, имя и отчество, даты жизни исследователя; название и иллюстрация открытия.

Примеры слайдов:

 Можайский Александр Фёдорович (1825 - 1890) изобретатель первого в мире самолёта 	 Менделеев Дмитрий Иванович (1834 - 1907) открыл периодический закон химических элементов, создал одноимённую таблицу 
 Сикорский Игорь Иванович (1889 - 1972) создал первый в мире вертолёт и бомбардировщик 	 Попов Александр Степанович (1859 - 1906) первый изобретатель РАДИО 
 Блинов Фёдор Абрамович (1827 - 1902) изобрёл первый в мире гусеничный трактор 	 Фёдоров Владимир Григорьевич (1874 - 1966) изобрёл и изготовил первый в мире автомат 

Список литературы

Основная литература:

1. Информационные технологии и управление предприятием [Электронный ресурс] / В.В. Баронов [и др.]. — Электрон. текстовые данные. — Саратов: Профобразование, 2017. — 327 с. — 978-5-4488-0086-3. — Режим доступа:

<http://www.iprbookshop.ru/63813.html>

2. Барский А.Б. Параллельные информационные технологии [Электронный ресурс] : учебное пособие / А.Б. Барский. — Электрон. текстовые данные. — Москва, Саратов: Интернет-Университет Информационных Технологий

(ИНТУИТ), Вузовское образование, 2017. — 503 с. — 978-5-4487-0087-3. —

Режим доступа: <http://www.iprbookshop.ru/67379.html>

3. Основы информационных технологий [Электронный ресурс] : учебное пособие / Г.И. Киреева [и др.]. — Электрон. текстовые данные. — Саратов: Профобразование, 2017. — 272 с. — 978-5-4488-0108-2. — Режим доступа:

<http://www.iprbookshop.ru/63942.html> **Дополнительная литература:**

1. Гохберг Г.С., Зафиевский А.В., Короткин А.А. Информационные технологии. ОИЦ «Академия», 2022.