Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Баламирзоев Назим Гидатович
Должность: Ректор
Дата подписания: 11.09.2025 13:36:02
Уникальный программный ключ:
5cf0d6f89e80f49a334f6a4ba58e91f3326b9926

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДАГЕСТАНСКИЙ ГОСУДАСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Учебно-методические указания

к выполнению практических работ №1-№4

по дисциплине «Введение в современные компьютерные технологии» для обучающихся по направлению подготовки бакалавров 01.03.02- «Прикладная математика и информатика», профилю «Системное программирование и компьютерные технологии».

Махачкала 2022г.

УДК 681.3.06(072)

Учебно-методические указания к выполнению практических работ №1-№4 по дисциплине «Введение в современные компьютерные технологии» для обучающихся по направлению подготовки бакалавров 01.03.02-«Прикладная математика и информатика», профилю «Системное программирование и компьютерные технологии».-Махачкала: ИПЦ ДГТУ, 2022г.-40с.

Учебно-методические указания подготовлены в соответствии с Государственным образовательным стандартом высшего образования.

Указания содержат описания лабораторных работ по темам:

Работа в среде текстового процессора MS Word.

Облачные технологии для передачи и обработки информации.

Создание интеллектуального чат-бота.

Технология Интернет вещей.

Каждая работа содержит теоретический материал и практическую часть.

Составители: ст. преподаватель кафедры «Прикладной математики и информатики» Алиосманова О.А, ст. преподаватель кафедры «Прикладной математики и информатики» Эседова Г.С., ст. преподаватель кафедры «Прикладной математики и информатики» Гаджимахадова Л.М.

Рецензент:

Зав. кафедрой «ПОВТ и АС» ДГТУ, к.э.н., доцент Т.Г. Айгумов.

Директор НИИ Региональных проблем информатизации ДГУНХ, профессор кафедры «ИТи ИБ», д.э.н. С.Э. Савзиханова.

Печатается согласно постановлению
Ученого совета Дагестанского государственного технического университета
от _____2022 г

Оглавление

Лабораторная работа № 1. Облачные технологии для передачи и обработки информации..... 4
Задание №1. ……..12
Задание №2 по Google Документам 13
Задание №3 по Google Таблицам 16
Задание№ 4 по Google Формам. Создание опроса с помощью Google Форм 17
Задание……..... 20
Лабораторная работа № 2. Создание интеллектуального чат-бота..... 20
Лабораторная работа № 3. Технология Интернет вещей. 25
Индивидуальные задания. 36
Лабораторная работа № 4. Введение в машинное обучение. ……..36
Задание…… 40
Список литературы. 40

Лабораторная работа № 1. Облачные технологии для передачи и обработки информации.

***Цель работы**: изучение возможностей интернет-сервисов по созданию, совместному редактированию и опубликованию документов.*

Теоретический материал

Сегодня пользователи всѐ чаще отдают предпочтение веб-сервисам, освобождая свой компьютер от лишних программ, функции которых уже реализованы и доступны онлайн. Многие используют online сервисы как в частной жизни, так в деловой и профессиональной деятельности. К тому же большинство онлайновых услуг предоставляются бесплатно, что только дополняет их преимущество.

Любой сервис онлайн отличает в первую очередь то, что работа в нем осуществляется непосредственно через браузер и не требует для своего функционирования дополнительных программ. По этой причине каждый онлайн сервис вне зависимости от его предназначения, имеет ряд преимуществ:

– не нужно устанавливать на компьютер программное обеспечение, достаточно просто открыть окно браузера;

– отпадает проблема совместимости между той или иной программой и операционной системой на компьютере пользователя;

– нет необходимости отслеживать обновления, сервис онлайн - это всегда актуальная версия программы;

– online сервис доступен всегда и везде, он не привязан к конкретному ПК.

Сервисы Интернет можно классифицировать также по выполняемым ими функциям как:

– Сервисы для хранения закладок;
– Сервисы для хранения мультимедийных ресурсов;
– Создание и редактирование документов;
– Социальные геосервисы;
– Социальные сети;
– Социальные поисковые системы.

Количество сервисов в сети растет с каждым днем, появляются все новые и новые сервисы, облегчающие жизнь человека, это нашло отражение в современной концепции развития всемирной паутины Веб 2.0.Веб 2.0 – это второе поколение сетевых сервисов, действующих в Интернете. В отличие от сетевых сервисов первого поколения Веб 2.0 ориентированы на совместную работу пользователей, обмен информацией, а также работу с массовыми публикациями.

Многие корпорации объединяют под своей эгидой несколько сервисов, что облегчает работу с ними, так как достаточно одного аккаунта для работы с каждым изних. Наиболее популярными при этом являются Google и Yandex. В таблице 1 приведены названия и ссылки на некоторые из сервисов этих корпораций, сгруппированные по типу.

Табл. 1. Онлайн-сервисы корпораций Googleи Yandex

Тип сервиса	Yandex	Google
Поиск информации	**Яндекс Поиск** https://yandex.ru/, https://ya.ru/	**Google Поиск** https://www.google.ru/
Хранение данных	**Яндекс.Диск** https://disk.yandex.ru	**Google Drive** https://drive.google.com/drive
Карты	**Яндекс.Карты** https://yandex.ru/maps	**Google Maps** https://www.google.com/maps
Почта	**Яндекс.Почта** https://mail.yandex.ru	**Gmail** https://mail.google.com/mail
Вопросы	**Яндекс.Знатоки** https://yandex.ru/znatoki/	**Вопросы и ответы** Сервисзакрыт

Календарь	**Яндес.Календарь** https://calendar.yandex.ru	**Календарь Google** https://calendar.google.com/
Сайты	**Яндекс.Народ** Хостинг передан веб-сервису uCoz	**Google Sites** https://sites.google.com
Документы	отсутствует	**Google Docs** https://docs.google.com
Формы	**Яндекс.Формы** https://forms.yandex.ru	**Google Формы** https://docs.google.com/forms
Фото сервис	**Яндекс.Фотки** Сервис закрыт (фотографии перенесены на Яндекс.Диск)	**Google Фото** https://photos.google.com/
Видео сервис	**Яндекс.Видео** https://yandex.ru/portal/video	**YouTube** https://www.youtube.com/
Переводчик	**Яндекс.Переводчик** https://translate.yandex.ru/	**Google Переводчик** https://translate.google.com/

В данной лабораторной работе на примере Google Docs будут рассмотрены возможности онлайн-сервисов по созданию, редактированию и передаче документов.

Сервисы Google позволяют:

–создавать в Интернете документы, электронные таблицы, анкеты и презентации, а затем просматривать и редактировать их с любого компьютера, подключенного к Интернету;

–систематизировать документы с помощью папок в хранилище Диск Google;

–экспортировать созданные файлы на ваш компьютер;

–загружать уже готовые файлы с вашего компьютера и размещать их в хранилище Диск Google;

–опубликовать документ в виде веб-страницы или разместить в своем блоге;

–предоставлять доступ к редактированию или просмотру своих документов другим пользователям;

–в режиме реального времени редактировать документ одновременно несколькимлюдям;

–совместно просматривать презентации;

–обмениваться важными документами, таблицами и презентациями между сотрудниками компании или коллектива.

Работа в онлайн текстовым редакторeGoogle Документы

Регистрация и авторизация

Чтобы получить доступ к сервису Google Docs, необходимо иметь профиль в системе Google. Наличии аккаунта дает доступ ко всем инструментам Google, в том числе сервису Docs. Алгоритм регистрации и авторизации следующий:

1. Регистрируемся в Google и подтверждаем свой профиль(номер телефона, E-mail).
2. Авторизуемся в службах Google путем ввода личного Логина и секретного пароля.
3. Переходим по ссылке к Google: http://docs.google.com/

Второй способ попасть в Документы – кликнуть на странице Google по значку ⠿ в правом верхнем углу, раскрыть полный список служб (нажать «Еще») и выбрать здесь «Документы».

Создание нового документа/файла

При первом запуске Google Документов с нового аккаунтаоткроется пустое окно, т.к. документы еще не созданы, не загружены. Если же для работы использовать старый аккаунт Google, то отобразится список недавних файлов, открытых ранее (рис. 1).

1. Чтобы создать новый файл в категории «Документ» (аналог файла MS Word), необходимо нажать на значок плюс (+), размещенный в правом нижнем углу.

2. Чтобы присвоить новому файлу имя/название, необходимо поставить курсор на название файла «Новый документ» в левом верхнем углу, выше панели инструментов и изменить на любое другое. При нормальном соединении с Интернетом все изменения в файлах сохраняются автоматически на Google Диске, о чем свидетельствует сообщение

«Все изменения сохранены на Диске» справа на панели меню.

3. При создании нового файла система автоматически создает новый документ на Google Диске профиля. Кроме того этот файл помещается в список «Недавние документы», который отображается при входе в службу Google Docs.

Редактирование документа

Интерфейссервиса Google Docs идентичен Microsoft Office (Word, Excel, PowerPoint). В случае затруднений при поиске той или иной функции можно воспользоваться справкой: меню «Справка» - пункт «Справка – GoogleДокументы».

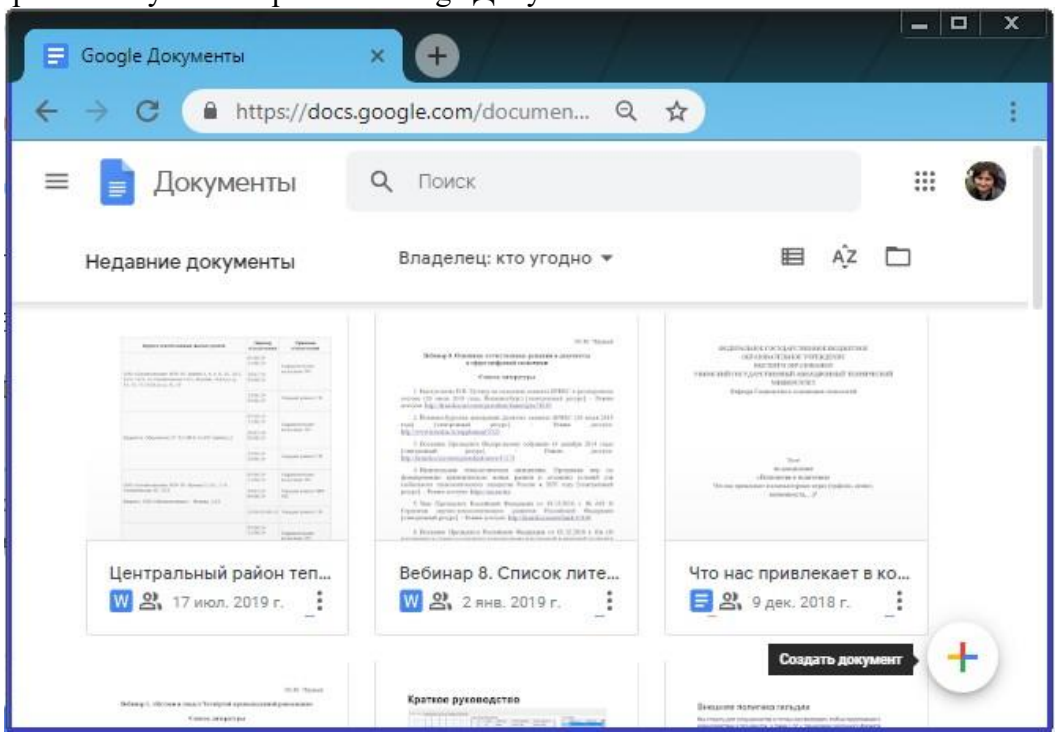


Рис.1. Окно работы с GoogleДокументами.

Настройки доступа

Чтобы другой человек открыл наш документ и смог вносить в него правки, нужно настроить систему доступа к файлу. Для этого нужно в верхнем правом углу нажать на кнопку «Настройки доступа», откроется окно «Совместный доступ» (рис. 2).

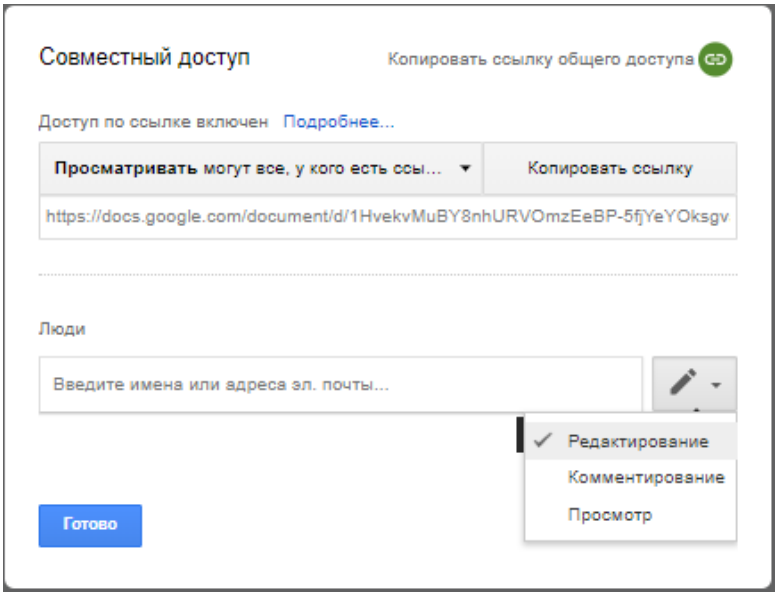


Рис. 2. Окно «Совместный доступ»

Можно дать индивидуальный доступ конкретным пользователям, имеющим Google-аккаунт, и/или включить общий доступ для всех, у кого есть ссылка:

1. Чтобы включить доступ по ссылке, нужно в правом верхнем углу окна

«Совместный доступ» нажать на кнопку . Ссылка будет сгенерирована автоматически. Нажатием кнопки «Копировать» она будет помещена в буфер обмена. Теперь эту ссылку можно отправить в личном сообщении, опубликовать на форуме или сайте и т.д.

2. Чтобы предоставить индивидуальный доступ конкретным пользователям, необходимо в окне «Совместный доступ» в поле «Люди» перечислить их адреса Gmail. Можно добавить описание, а также оповестить добавленных пользователей через email – для этого нужно поставить галочку в поле «Оповестить пользователей».

Возможны следующие варианты доступа:

–Просматривать – пользователи смогут просматривать данный файл;

–Комментировать – пользователи смогут оставлять комментарии через опцию комментирования;

–Редактировать – пользователи смогут вносить свои правки в документ, которые запишутся в «Истории изменений».

Вы можете в любое время закрыть доступ к файлу, а также запретить другим пользователям изменять объект и настройки доступа к нему.

История изменений файла

Чтобы узнать, какие изменения вносились пользователями в документ Google, нужно перейти в меню «Файл» и выбрать пункт «История версий», затем «Смотреть историю версий». Справа откроется окно «История версий» со списком производимых пользвателями правок. Выбрав версию на панели справа, можно увидеть, кто именно вносил изменения в файл и каким цветом выделены правки каждого пользователя.

Чтобы присвоить версии название, нажмите на значок ⋮ и из выпадающего списка выбрать «Указать название версии».

Чтобы вернуться к текущей версии файла, нажмите на значок ← в левом верхнем углу экрана.

История версий доступна только пользователям с правом на редактирование файла.

Комментирование документа

При работе с документами Google можно взаимодействовать с соавторами: добавлять, редактировать и удалять комментарии, а также отвечать на них.

Чтобы оставить свой комментарий необходимо:

1. Получить доступ к возможности комментировать файл, а не только его просматривать. Автору документа опция комментирования доступна по умолчанию.

2. Выделить в тексте нужный фрагмент и справа от этой строки нажать на значок «Добавить комментарий».

3. В открывшемся блоке нужно ввести свой комментарий и нажать «Комментировать».

4. Кроме того, мы также можем ответить на комментарии других пользователей, совместно работающих с этим файлом. Если кликнуть на комментарий, под ним появится пустое поле «Введите ответ…», а также будет доступна кнопка «Вопрос решен».

5. Если же нужно изменить или удалить свой комментарий, необходимо нажать на пиктограмму справа от кнопки «Вопрос решен» и выбрать одну из опций «Изменить» или «Удалить».

6. Посмотреть полный список комментариев к файлу, а также историю ответов на комментарии, можно с помощью кнопки «Комментарии» в правом верхнем углу окна, слева от кнопки «Настройки Доступа».

Скачивание файла Google Docs на компьютер

Все файлы Google Docs могут быть загружены на компьютер. Сделать это можно, например, через интерфейс самого документа. В файле в меню «Файл» необходимо выбрать «Скачать как…» и указать формат будущего файла. Доступные типы форматов:

- Microsoft Word (DOCX);
- Формат OpenDocument (ODT);
- Текст в формате RTF;
- Документ PDF;

– Текст в формате TXT;
– Веб-страница (HTML, ZIP-архив);
– EPUB.

Работа с табличнымонлайн процессором Google Таблицы

Google Таблицы – это онлайн-приложение, с помощью которого можно создавать и форматировать таблицы, а также работать над ними совместно с другими пользователями.

Все операции в Google Таблицах по созданию, настройкам доступа, комментированию, просмотру истории изменений, скачиванию аналогичны операциям с Google Документами.

Чтобы начать работу с Google Таблицами, необходимо в окне работы с документами (рис. 1) нажать на пиктограмму в левом верхнем углу и из списка выбрать пункт Таблицы

Интерфейс Таблиц напоминает нам Microsoft Excel с тем отличием, что здесь присутствуют опции «Комментарии», «Скачать файл», «История изменений», «Настройки доступа», а также прочие функции, характерные для Google Docs.

В Google Таблицах также доступны:

– автоматические расчеты по заданным формулам;
– графики, наглядные диаграммы, фильтры и функции;
– все арифметические операции, производимые в режиме онлайн;
– настройки отдельных ячеек, варианты вывода данных;
– условное форматирование;
– опции копирования, вырезания, вставки, печати, импорта/экспорта данных и т.д.

Дополнительную информацию о Google Таблицах можно получить, воспользовавшись справкой (меню «Справка» - пункт «Справка – Справка по таблицам».).

Работа с Google Формами

С помощью Google Форм можно создавать онлайн-опросы и тесты, а также отправлять их другим пользователям.

Чтобы начать работу с Google Формами, необходимо в окне работы с документами (рис. 1) нажать на пиктограмму в левом верхнем углу и из списка выбрать пункт Формы

Создание опроса с помощью Google Форм

Чтобы создать новую форму, нужно перейти на страницу forms.google.com и навести указатель мыши на знак плюса (+) в правом нижнем углу окна. Для создания чистой формы с нуля необходимо нажать на значок , для выбора одного изсуществующих шаблонов – . Откроется форма нового опроса (рис. 3) или теста.Создаваемая с нуля форма по умолчанию является опросом, т.е. служит для сбора статистической информации, общественного мнения и т.д., вопросы в ней не имеют правильных или неправильных ответов и не предполагают начисления баллов.

В форму можно добавлять вопросы, видео, изображения, а также разделы – они позволяют структурировать данные.

Добавление и редактирование вопроса

Чтобы добавить новый вопрос, нажмите на значок , справа от вопроса выберитетип. В формах и тестах можно собирать ответы разных типов:

1. Свободный ответ:
   – Текст (строка) – краткий ответ, респонденты отвечают в нескольких словах;
   – Текст (абзац) – развернутый ответ, респонденты дают ответы длиной в несколько строк или абзацев.

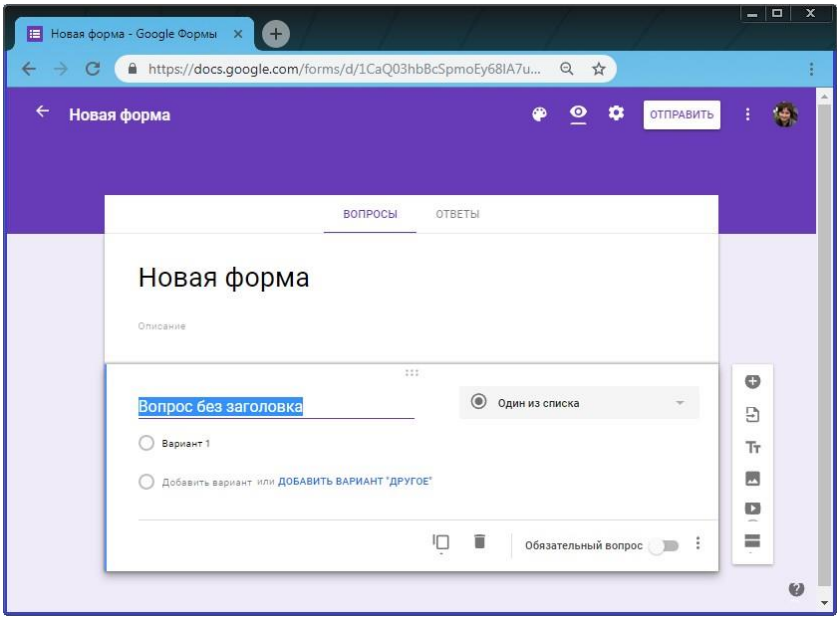


Рис. 3. Создание опроса с помощью Google Форм

2. Выбор из нескольких вариантов:
   – Один из списка – респондентам предлагается несколько вариантов ответа, можно выбрать только один ответ. При добавлении варианта «Другое», пользователи смогут ввести короткий ответ.
   – Несколько из списка – респондентам предлагается несколько вариантов ответа, можно выбрать несколько ответов. При добавлении варианта «Другое», пользователи смогут ввести короткий ответ.
   – Раскрывающийся список – респондентам предлагается несколько вариантов ответа, можно выбрать только один ответ.
3. Загрузка файлов (Загрузка файлов) – в качестве ответа на вопрос респонденты могут загрузить файл, при этом вход в аккаунт Google станет для нихобязательным. Загруженные файлы помещаются в хранилище Google Диска, принадлежащего владельцу опроса.
4. Выбор из сетки:
   – Шкала –респонденты дают оценку вопроса по шкале, которая может начинаться с 0 или 1 и заканчиваться целым числом от 2 до 10;
   – Сетка (множественный выбор) – респонденты выбирают один вариант из строки;
   – Сетка флажков – респонденты выбирают один или несколько предложенных вариантов из строки.
5. Выбор даты и времени:
   – Дата – респонденты могут указать любую дату;
   – Время – респонденты могут указать время или продолжительность.

Добавьте варианты ответа на вопрос. Если респонденты должны обязательно нанего ответить, включите параметр Обязательный вопрос .

В вопросы с выбором одного или нескольких ответов можно ***добавлять изображения***. Для этого необходимо нажать на вопрос или ответ, справа от него кликнуть на значок , выбрать или загрузить изображение и нажать «Выбрать».

Изменение порядка вопросов и ответов. Можно сделать так, чтобы респонденты видели вопросы и ответы в разном порядке:

1. Чтобы перемешать вопросы, нажмите на значок в верхней части страницы,

выберите вкладку «Презентация» и нажмите «Перемешать вопросы».

2. Чтобы респонденты видели варианты ответов в разном порядке, нажмите на вопрос, ответы которого нужно перемешать, кликните на значок в правом нижнем углу карточки с вопросом и выберите пункт «Перемешать ответы».

Чтобы ***удалить*** вопрос или изображение, его необходимо выделить и нажать на значок .

Добавление и редактирование раздела

Чтобы работать с формой было удобнее, можно разбить ее на разделы. Для этого нужно нажать на значок "Добавить раздел" и указать название раздела.

Если в форме имеется несколько разделов, можно изменить их порядок. Для этого нужно нажать на значок в правом верхнем углу любого раздела и выбрать «Переместить раздел». Чтобы переместить раздел, нажмите на стрелку вверх или вниз

Чтобы удалить раздел, нажмите на заголовок раздела, кликните на значок и выберите «Удалить раздел».

Отмена действия

Чтобы отменить недавнее изменение в форме, нажмите на значок в правом верхнем углу формы и выберите Отменить .

Создание и оценка тестов с помощью Google Форм

С помощью Google Форм можно создать тест, отправить его респондентам и оценить.

Создание теста на основе формы

Создать тест можно на основе существующей или новой формы. Для этого нажмите на значок настроек в правом верхнем углу экрана, в окне «Настройки» перейдите на вкладку «Тесты» (рис. 4), включите параметр Тест , нажмите «Сохранить».

После того, как опрос преобразован в тест, за ответы на любые типы вопросов, кроме множественного выбора (сетки), можно назначить количество баллов.

Типы вопросов, для которых баллы начисляются автоматически:

– один из списка;
– несколько из списка;
– раскрывающийся список;
– текст (строка).

Для назначения количества баллов нужно выделить вопрос и нажать на ОТВЕТЫ в левом нижнем углу экрана. В открывшемся окне нужно отметить правильные ответы и в правом верхнем углу указать, во сколько баллов вы оцениваете вопрос.

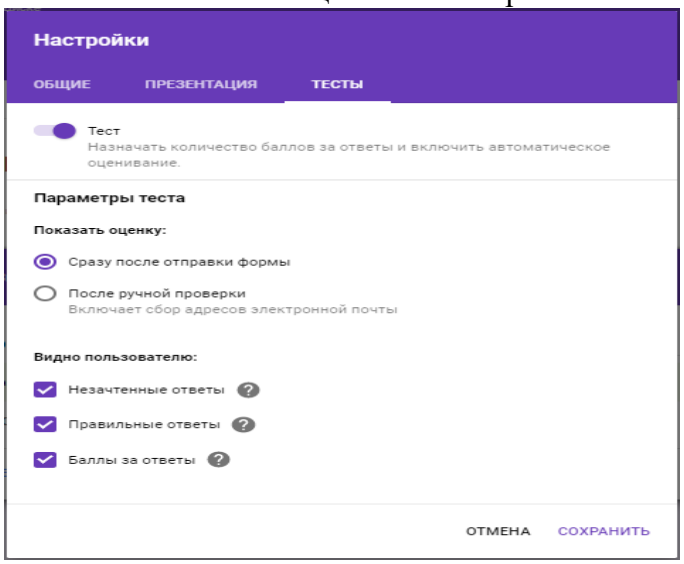


Рис. 4. Преобразование опроса в тест

Можно выбрать, что будут видеть респонденты во время и после выполнения теста:

– незачтенные ответы (на какие вопросы даны неверные ответы);

– правильные ответы (показывать правильный ответ для каждого вопроса после выставления оценок);

– баллы за ответы (показывать общее количество баллов и количество баллов за каждый вопрос).

Чтобы изменить параметры, нажмите на значок настроек в правом верхнем углу экрана, перейдите на вкладку «Тесты» (рис. 4) и отметьте флажками нужные параметры в разделе "Видно пользователю".

Оценка ответов на тесты

Вы можете просмотреть автоматическую сводку всех ответов на тест, которая включает:

– вопросы, на которые часто даются неправильные ответы;

– диаграммы, показывающие процент правильных ответов;

– диапазон баллов, а также их среднее и медианное значения.

Чтобы просмотреть сводку результатов, перейдите на вкладку «Ответы» в верхней части формы (рис. 3), выберите «Сводка».Чтобы просмотреть ответы определенного респондента, нажмите «Отдельный пользователь».

Для организации обатной связи с респондентами, можно включить опцию сбора адресов электронной почты: Нажмите на значок настроек в правом верхнем углу экрана и на вкладке«Общие» установите флажок Собирать адреса электронной почты .

Если вы сохраняете адреса электронной почты респондентов, у вас есть возможность оценивать ответы и оставлять отзывы на них (не забывайте сохранять изменения):

1. Нажмите «Ответы» в верхней части формы, выберите «Отдельный пользователь».
2. Чтобы перемещаться между ответами, нажимайте на стрелки назад и вперед
3. Выберите вопрос, который нужно оценить.
   – В правом верхнем углу укажите количество баллов.
   – Под ответом нажмите «Добавить личный комментарий».
   – Оставьте комментарий и нажмите «Сохранить».
4. Чтобы сохранить изменения, нажмите «Сохранить».
5. После этого вы можете отправить результаты по электронной почте.

После проверки результатов вы можете отправить их респондентам по электронной почте сразу, либо в другое время. Для этого в настройках на вкладке «Тесты» (рис. 3) в разделе «Показать оценку» необходимо отметить подходящий пункт:

– Сразу после отправки формы – респонденты будут получать результаты сразу после выполнения теста.

– После ручной проверки – результаты можно будет отправить позже по электронной почте.

Для отправки результатов по электронной почте на вкладке «ответы» - «Сводка» найдите раздел «Баллы», нажмите «Опубликовать баллы», выберите тех, кому вы хотите отправить результаты, нажмите «Опубликовать и сообщить по эл. Почте».

Отправка формы респондентам

Готовую форму можно отправить респондентам по электронной почте или через социальные сети. Кроме того, ее можно встроить в веб-страницу.

Чтобы отправить форму по электронной почте, нажмите кнопку «Отправить» в правом верхнем углу экрана. В открывшемся окне на вкладке укажите адреса, накоторые хотите отправить форму, а также тему и текст письма. Нажмите «Отправить».

Чтобы получить ссылку на форму, нажмите кнопку «Отправить» в правом верхнем углу экрана.В открывшемся окне нажмите на вкладку. Скопируйте ссылку на форму, теперь ее можно отправить через чат, по электронной почте и т.д.

Сохранение ответов респондентов в Google Таблице

Хранить ответы респондентов можно как в самой форме, так и в таблице Google.

1. Перейдите на вкладку «Ответы» и выберите «Сводка» в левой верхней части экрана.
2. Нажмите на значок ⁝ в правой верхней части экрана и выберите «Сохранять ответы».
3. Выберите один из вариантов:
   – Новая таблица - в Google Таблицах будет создан файл для ответов.
   – Существующая таблица - ответы будут храниться в уже имеющемся файле.
4. Нажмите «Создать» или «Выбрать».

Файл с ответами появится в Google Таблицах (в приложении и веб-версии), а такжена Google Диске.

Задание №1.

Для выполнения заданий необходимо разделиться на команды по 2-3 человека. У каждого студента должен быть зарегистрирован аккаунт на Gmail.

Задание 1.

1. Создайте новый документ в GoogleDocs под одним из аккаунтов. Откройте документ Word, созданный в ходе выполнения лабораторной работы №5, скопируйте из него текст и вставьте в документ GoogleDocs.
2. Предоставьте индивидуальный доступ с правами редактирования документа GoogleDocs для аккаунта другого студента команды. Оповестите его через электронную почту.
3. Включите доступ по ссылке с правами просмотра. Перешлите ссылку другим студентам группы через любой месенджер, социальную сеть или email.
4. Выйдите с текущего аккаунта GoogleDocs и авторизуйтесь под учетной записью студента, которому предоставили доступ с правами редактирования. Откройте документ GoogleDocs через оповестительное письмо.
5. Скопируйте из документа Word рисунок и вставьте его в документ GoogleDocs. Добавьтеверхний колонтитул и нумерацию страниц. Добавьте комментарий к любому абзацу текста.
6. Снова перейдите к учетной записи создателя документа и откройте документ GoogleDocs. Ответьте на комментарий. Просмотрите историю версий.
7. Скачайте файл Google Docs на компьютер.

Задание 2.

1. Находясь в Google Документах, перейдите в Google Таблицы. Создайте новую Google таблицу и загрузите в нее Excelфайл, созданный в ходе выполнения лабораторной работы №5. Убедитесь, что все формулы и условное формаирование были также перенесены.
2. Выполните п.3 и п.4 задания 1 для Google Таблиц.
3. Измените пороговое значение. Измените цвета заливки в правилах условного форматирования. Добавьте комментарий к какой-нибудь ячейке.
4. Выполните п.6 задания 1 для Google Таблиц.

Задание 3.

1. Находясь в Google Таблицах, перейдите в Google Формы. Создайте опрос из не менее чем трех вопросов разных типов по тематике своего варианта. Добавьте изображения в один из вопросов или в варианты ответов к нему. Отправьте опрос респондентам с помощью ссылки (не менее двух).После получения ответов сохраните их в Google Таблицах.
2. Создайте тест из не менее чем трех вопросов разных типов по тематике своего варианта, отметьте правильные варианты ответов и укажите количество баллов за каждый вопрос. Добавьте обязательный вопрос типа «Текст (строка)», в котором респондент должен указать свою фамилию и имя, и переместите его в начало теста. Отключитевозможность многократного прохождения теста одним и тем же респондентом. Отправьте тест респондентам с помощью электронной почты (не менее двух респондентов). После получения ответов просмотрите сводку, результаты по каждому вопросу и по каждому респонденту.

Задание №2 по Google Документам

1. Создадим новый документ в GoogleDocs. Для этого откроем браузер и введем https://www.google.ru. После прохождения авторизации под своим аккаунтом Google нажмем на значок в правом верхнем углу и из списка сервисов выберем «Документы». В открывшемся окне (см. рис. 1) нажмем на значок плюса (+) в правом нижнем углу.

2. Дадим название нашему документу, для этого в левом верхнем углу кликнемна поле с текущим названием «Новый документ» и введем вместо него текст «Задание 1».

3. Откроем документ Word, созданный в ходе выполнения лабораторной работы №5, и с помощью сочетаний клавиш Ctrl+Cи Ctrl+Vскопируем из него текст в наш документ Google.

4. Предоставим индивидуальный доступ с правами редактирования для другого аккаунта. Для этого в правом верхнем углу окна нажмем на кнопку «Настройки доступа» и в открывшемся окне «Совместный доступ» (рис. 5) в строке «Люди»укажем Gmail-адрес пользователя, которму мы хотим дать доступ. Нажмем на значок справа от сроки и выберем пункт «Редактирование». Заполним поле «Описание» под строкой «Люди». Включим опцию «Оповестить пользователей» и нажмем кнопку «Отправить».

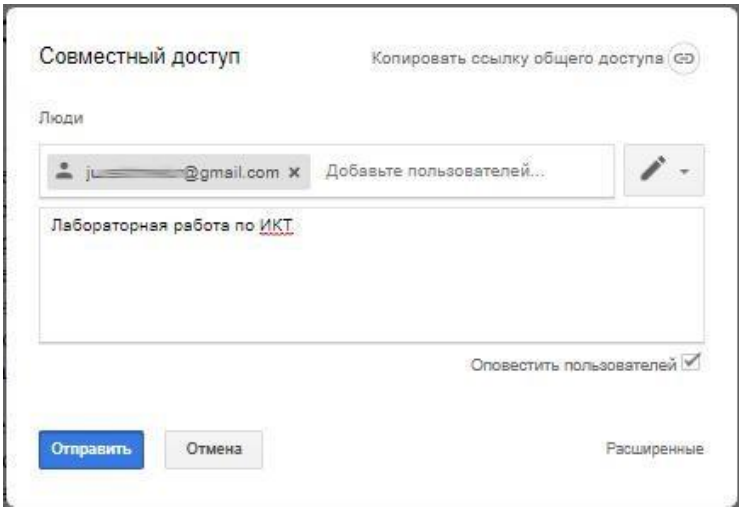


Рис. 5. Предоставление индивидуального доступа через окно «Совместный доступ»

5. Включим доступ по ссылке с правами просмотра. Для этого в окне «Совместный доступ» нажмем на значок в правом верхнем углу. В настройках доступа из выпадающего списка выберем пункт «Просматривать могут все, у кого есть ссылка» (рис. 6). Далее сгенерированную автоматически ссылку копируем, нажав на кнопку «Копировать ссылку» и отправим другим пользвателям через любой чат.

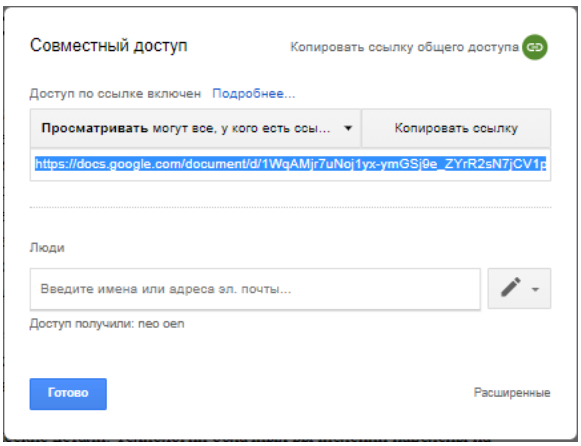


Рис. 6. Предоставление доступа по ссылке через окно «Совместный доступ»

6. В результате действий, описанных в п.4, другому пользователю было отправлено оповестительное письмо на ящик Gmail (рис. 7). С помощью него пользователь может перейти к редактированию нашего документа. Для этого в письме нужно нажать кнопку «Открыть в Документах».

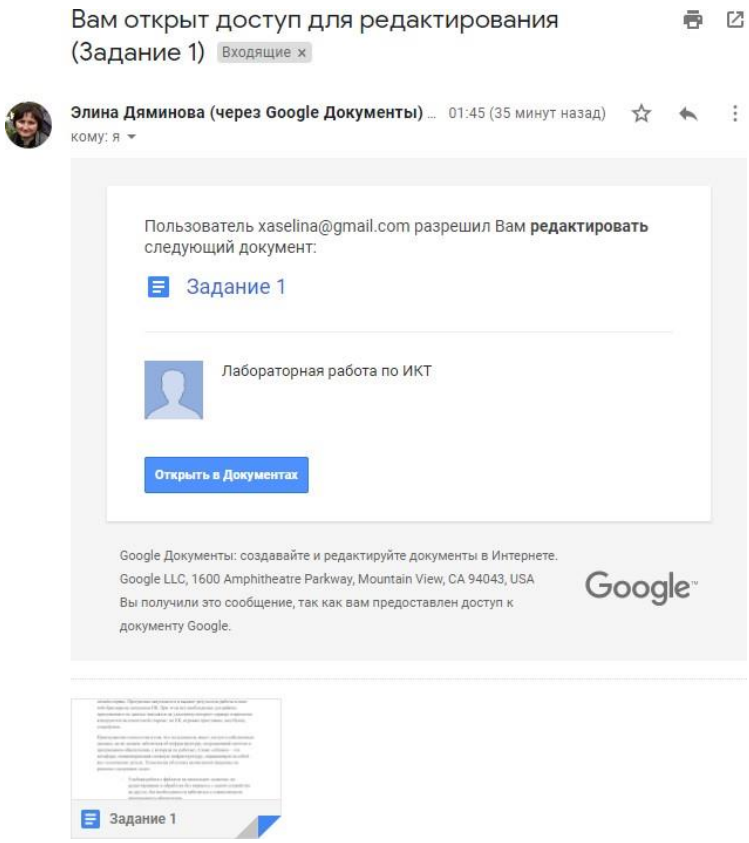


Рис. 7. Письмо о предоставлении доступа на редактирование Google документа

7. Далее с аккаунта второго пользователя отредактируем документ и добавим комментарий:

7.1. Скопирем из документа Word рисунок и вставьте его в текст документаGoogleDocs с помощью комбинации клавиш Ctrl+V.

7.2. Добавим верхний колонтитул через меню «Вставка» ➔ «Нумерация и верхний колонтитул»➔ «Верхний колонтитул». Укажем в тексте колонтитула номер группы и свои фамилии.

7.3. Добавим нумерацию страницчерез меню «Вставка» ➔ «Нумерация и верхний колонтитул»➔ «Номер страницы», далее выберем номер в правом нижнем углу страницы.

7.4. Добавим комментарий к фрагменту текста.Выделим в тексте нужный фрагмент и справа от этой строки нажмем на значок . В открывшемся блоке введем комментарийи нажмем «Комментировать» (рис. 8).

8. Вернемся к учетной записи создателя документа и откроем документ GoogleDocs. Чтобы ответить на комментарий, нажмем на него мышкой, в поле ответа введем текст, нажмем кнопку «Ответить», а затем «Вопрос решен» (рис. 9).

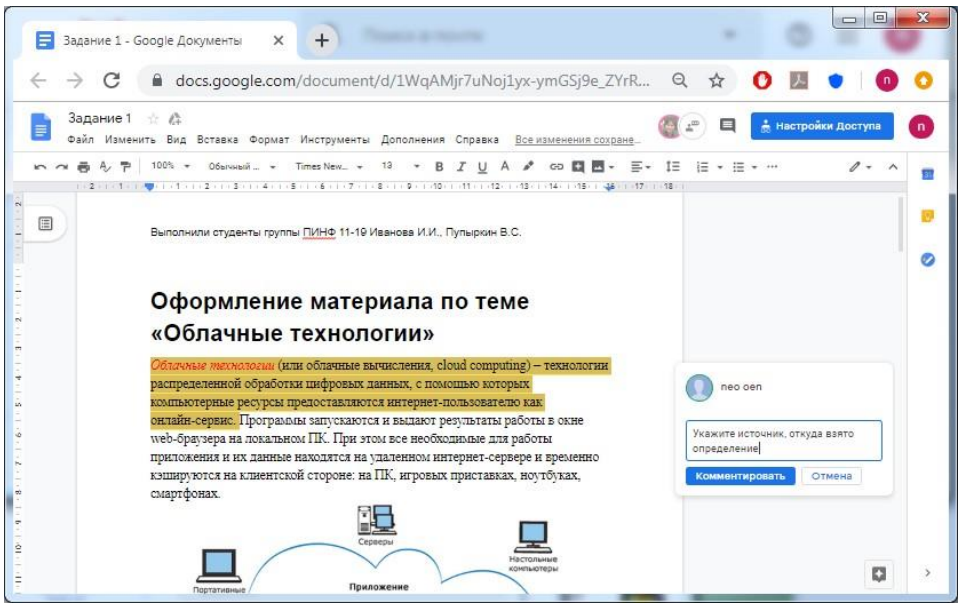


Рис. 8. Добавление комментария

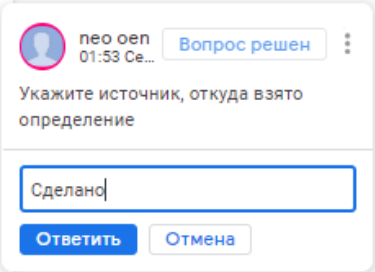


Рис. 9. Добавление ответа на комментарий

9. Просмотрим историю версий, выбрав в меню «Файл» ➔ «История версий» ➔ «Смотреть историю версий» (рис. 10). Как видно на рис. 10, изменения каждого пользователя подсвечиваются определенным цветом. В данном случае выделена последняя версия и колонтитул, который добавил второй пользователь, отмечен сиреневым цветом. Чтобы выйти из истории версий, нажмем в левом верхнем углу экрана.

10. Скачаем наш файл Google Docs на компьютер. Для этого выберем в меню «Файл» ➔ «Скачать» ➔ «Microsoft Word (DOCX)».

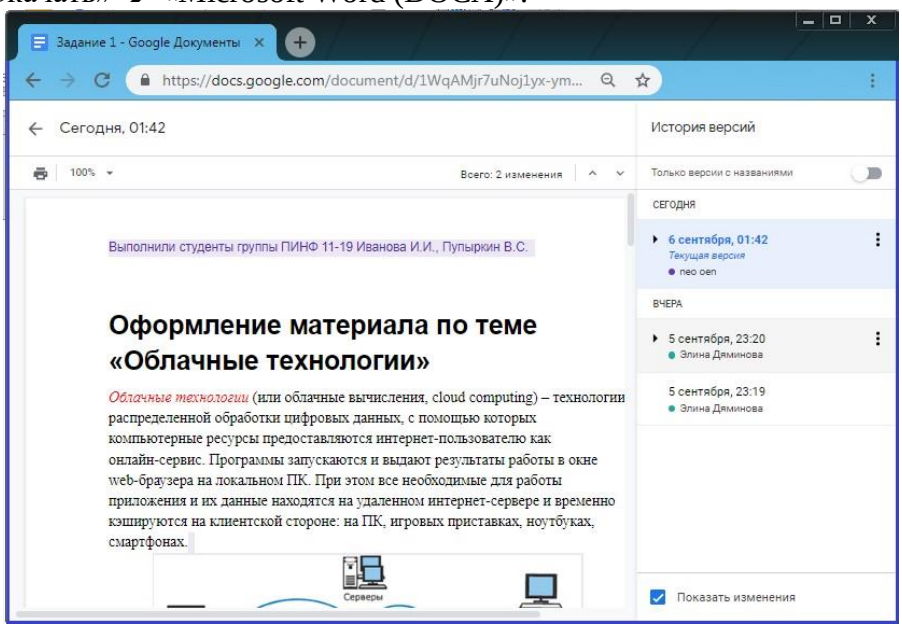


Рис. 10. Просмотр истории версий

Задание №3 по Google Таблицам

1. Вернемся к главной странице Google Документов, нажав на значок в левом верхнем углу окна. Затем нажмем на значок главного меню ≡ и выберем из списка Таблицы. Создадим новуютаблицу, нажав на значок плюса (+) в правом нижнем углу.

2. Загрузим в нашу новую талицу Excel файл, созданный в ходе выполнения лабораторной работы №5. Для этого в меню «Файл» выберем «Открыть». В появившемся окне перейдем на вкладку «Загрузка». Нажмем на кнопку «Выбрать файл на устройстве» и укажем путь к Excel файлу. После нажатия кнопки «Открыть» файл загрузится, название Excel файла также будет перенесено. Изменим название таблицы на «Задание 2».

3. Пощелкав по заполненым ячейкам столбцов «Сумма баллов» и «Результат», можно убедиться, что все формулы были перенесены корректно. Если же попробовать поменять баллы таким образом, чтобы их сумма стала меньше (либо наоборот, больше) порогового значения, то будет видно, что условное форматирование также работает.

4. Предоставим индивидуальный доступ другому пользователю и включим доступ по ссылке аналогичным образом, как это было реализовано для Google Документов (п. 4, 5 параграфа 5.1).

5. От лица другого пользователя откроем оповестительное письмо и перейдем к редактированию нашей Google таблицы:

5.1. В ячейке C19 введем новое пороговое значение – «55»;

5.2. Чтобы изменить цвета в правилах условного форматирования, выделим диапазон H3:H15, в меню «Формат» выберем «Условное форматирование». В открывшейся справа панели «Правила условного форматирования» нажмем на правило и изменим цвет текста и цвет заливки (рис. 11);

5.3. Добавим комментарий к ячейке C19, нажав на нее правой кнопкой мыши и выбрав из контекстного меню Комментарий. Введем текст комментария и нажмем «Комментировать» (рис. 11).

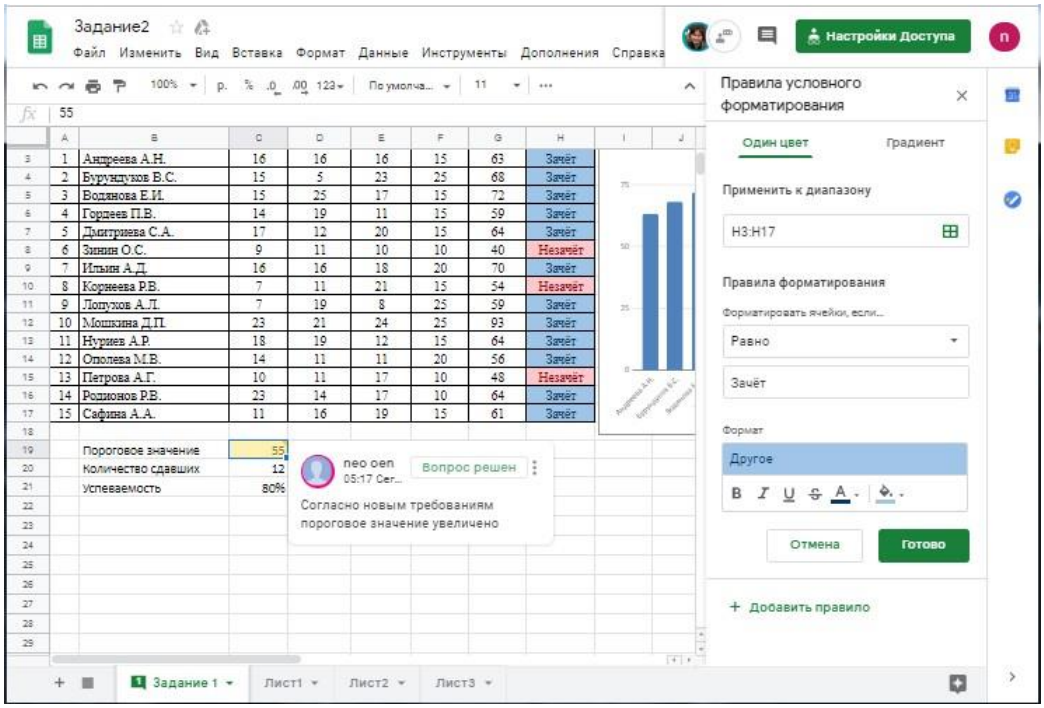


Рис. 11. Редактирование Googleтаблицы

6. Снова перейдем к учетной записи создателя таблицы и откроем документ Google Таблицы. Добавим ответ на комментарий и закроем его, а также просмотримисторию версий аналогичным образом, как это было реализовано для Google Документов (п. 8, 9)

Задание№ 4 по Google Формам. Создание опроса с помощью Google Форм

1. Вернемся к главной странице Google Таблиц, нажав на значок в левом верхнем углу окна. Затем нажмем на значок главного меню ≡ и выберем из списка Формы . Создадим новую форму, нажав на значок плюса (+) в правом нижнем углу.

2. Присвоим форме название: нажмем на надпись «Новая форма» в левом верхнем углу окна (см. рис. 3) и введем новое имя, например, «Опрос по онлайн- сервисам».

3. Добавим вопросы разных типов и варианты ответов к ним. Все вопросы будем считать необязательными для респондентов (рис. 12):

3.1. Пусть первый вопрос будет иметь тип раскрывающийся список – выберем соответствующий пункт из выпадающего списка справа. Введем формулировку вопроса вместо текста «Вопрос без заголовка» и добавим варианты ответа.

3.2. Добавим еще один вопрос, нажав кнопку справа. Допустим, это будет вопрос типа «Один из списка». Введем формулировку воспроса и варианты.

3.3. Третий вопрос сделаем типа «Несколько из списка» и добавим к вариантам ответов изображения, нажимая на значок справа от текста варианта ответа, указывая путь к файлу и загружая его. Также в этом вопросе добавим вариант «Другое», нажав на ДОБАВИТЬ ВАРИАНТ "ДРУГОЕ" .

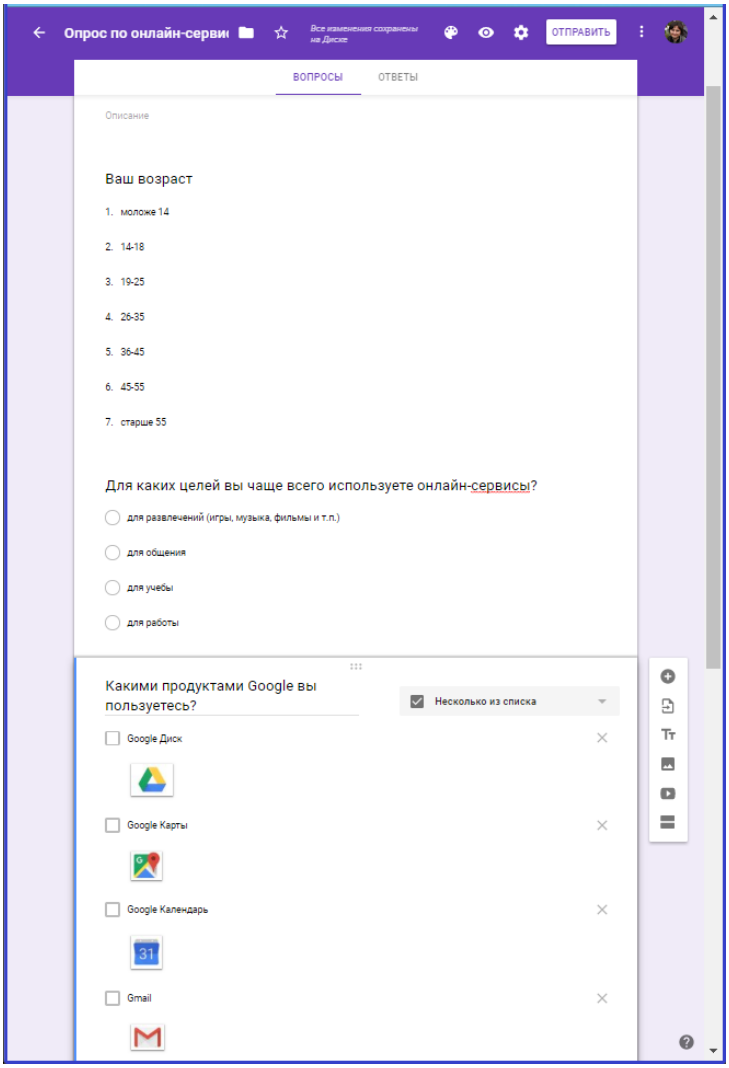


Рис. 12. Добавление вопросов

4. Чтобы отправить опрос респондентам с помощью ссылки, нажмем кнопку «Отправить» в правом верхнем углу экрана (см. рис 12).В открывшемся окненажмите на вкладку . Скопируем ссылку на форму и отправим респондентам любым удобным способом, например, по электронной почте (рис. 13).

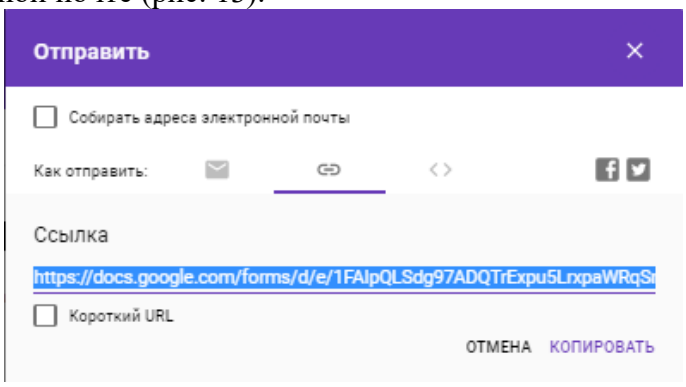


Рис. 13. Отправка формы респондентам с помощью ссылки

5. Чтобы ответы респондентов сохранялись вGoogle Таблицах, перейдем на вкладку «Ответы» и выберем «Сводка» в левой верхней части экрана.Нажмите на значокв правой верхней части экрана и выберем«Сохранять ответы». Выберем вариант «Новая таблица» и нажмем «Создать». Теперь среди наших Google Таблиц появилась таблица, в кот

рую автоматически будут помещаться ответы респондентов на данный опрос (рис.14).

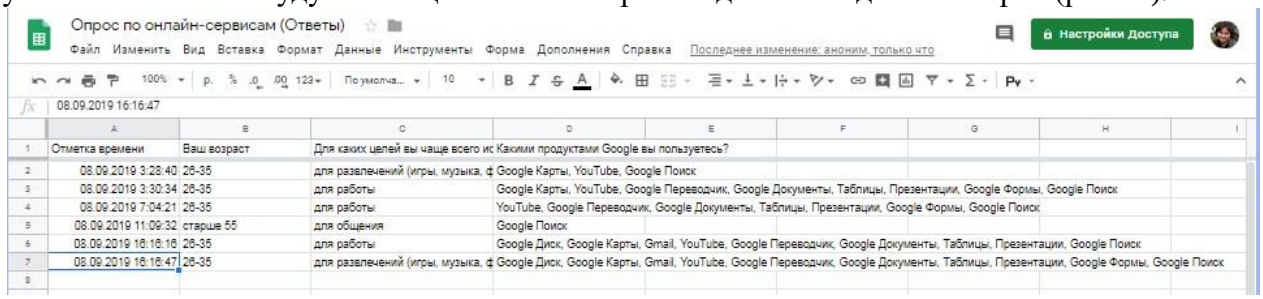


Рис. 14. Google Таблица с ответами на опрос

Создание теста с помощью Google Форм

1. Также как в п. 1 и п. 2 параграфа 5.3.1 создадим новую Google форму и дадим ей название, например, «Тест по GoogleDocs».

2. Чтобы преобразовать опрос в тест, нажмем на значок настроек в правом верхнем углу экрана, в окне «Настройки» перейдем на вкладку «Тесты» (см. рис. 4), включим параметр Тест , нажмем «Сохранить».Теперь за ответы вопросы можно назначить баллы.

3. Аналогично тому, как мы делали это при создании опроса, добавим вопросы и варианты ответов: один вопрос типа «Несколько из списка», один «Один из списка» и один «Текст (строка)».

4. Для назначения количества баллов выделим вопрос и нажмем на ОТВЕТЫ в левом нижнем углу. В открывшемся окне нужно отметим правильные ответы и в правом верхнем углу укажем, во сколько баллов вы оцениваете вопрос (рис. 15).

5. Добавим обязательный вопрос типа «Текст (строка)», в котором респондент должен указать свою фамилию и имя.Сделаем его обязательным, включив параметр Обязательный вопрос . Теперь для респондента этот вопрос будет отмечен звездочкой (*).

6. Чтобы переместить вопрос в начало теста, выделим его и наведем указатель мыши на значок сверху по центру вопроса. Когда указатель примет вид , зажмем левую кнопку мыши и перетащим вопрос наверх.

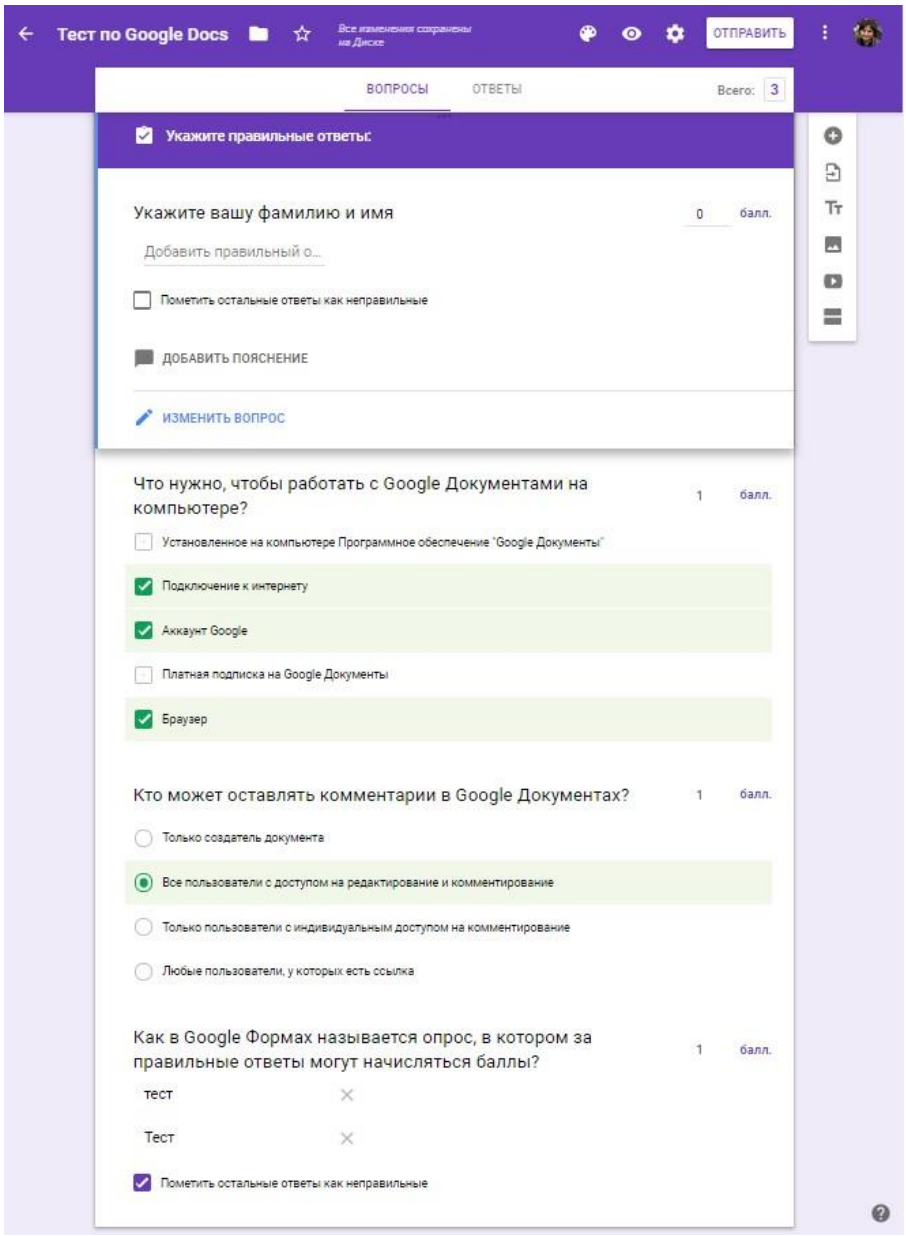


Рис. 15. Создание теста с помощью GoogleФорм

7. Чтобы отключить возможность многократного прохождения теста одним и тем же пользователем, нажмем на значок настроек в правом верхнем углу экрана, в открывшемся окне настроек перейдем на вкладку «Общие», поставим галочку на «Отправлять форму не более одного раза», и заодно на «Собирать адреса электронной почты», нажмем «Сохранить».

8. Чтобы отправить форму по электронной почте, нажмем кнопку «Отправить» в правом верхнем углу экрана. В открывшемся окне на вкладке укажем адресаэлектронной почты, а также тему и текст письма (рис. 16). Нажмем «Отправить».

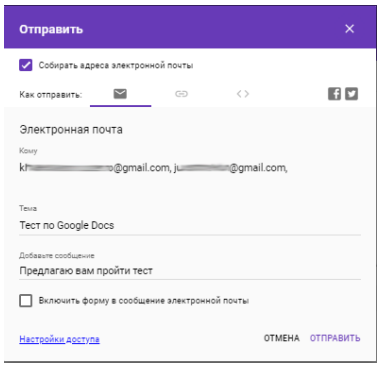


Рис. 16. Отправка формы респондентам по электронной почте

9. Для просмотра ответов респондентов перейдем на вкладку «Ответы» и выберем «Сводка» в левой верхней части экрана (рис. 17).

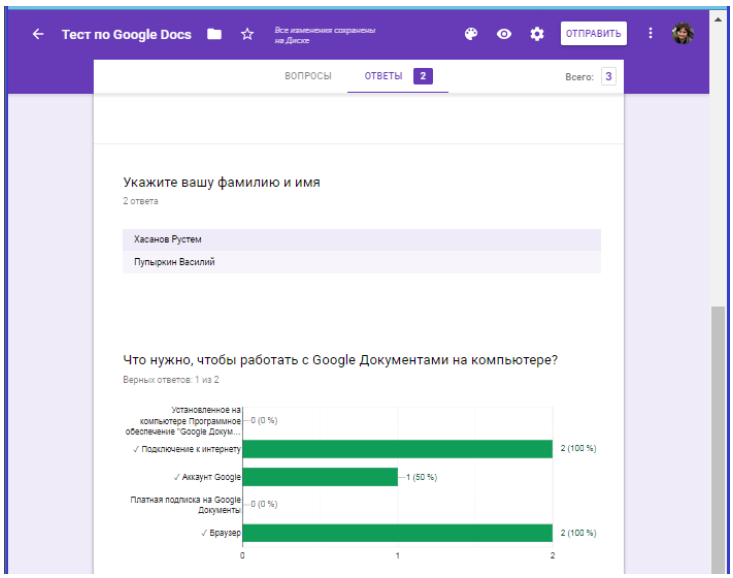


Рис. 17. Сводка по ответам на тест

Задание.

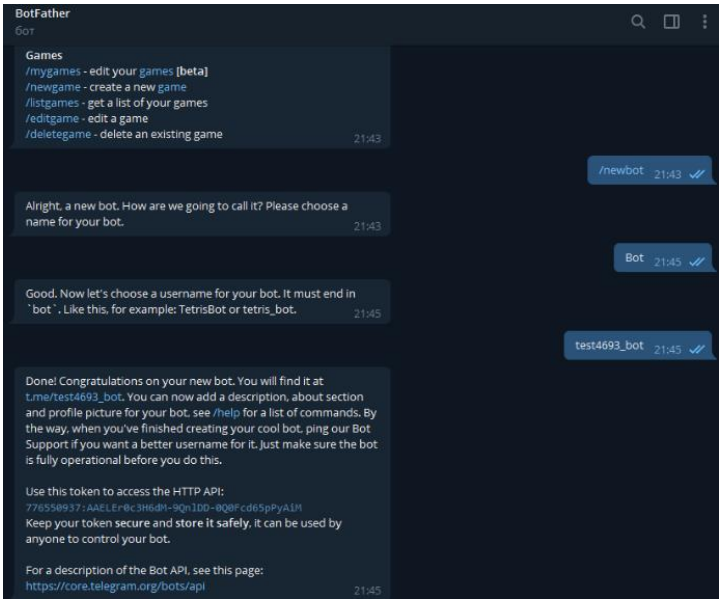
Результаты выполнения должны быть представлены в электронной форме (всего шесть документов GoogleDocs: GoogleДокумент и он же, сохраненный как документ Word, GoogleТаблица, опрос, результаты опроса в GoogleТаблице итест).

Лабораторная работа № 2. Создание интеллектуального чат-бота.

Цель работы: *изучение создания интеллектуального чат-бота на языке Pyton.*

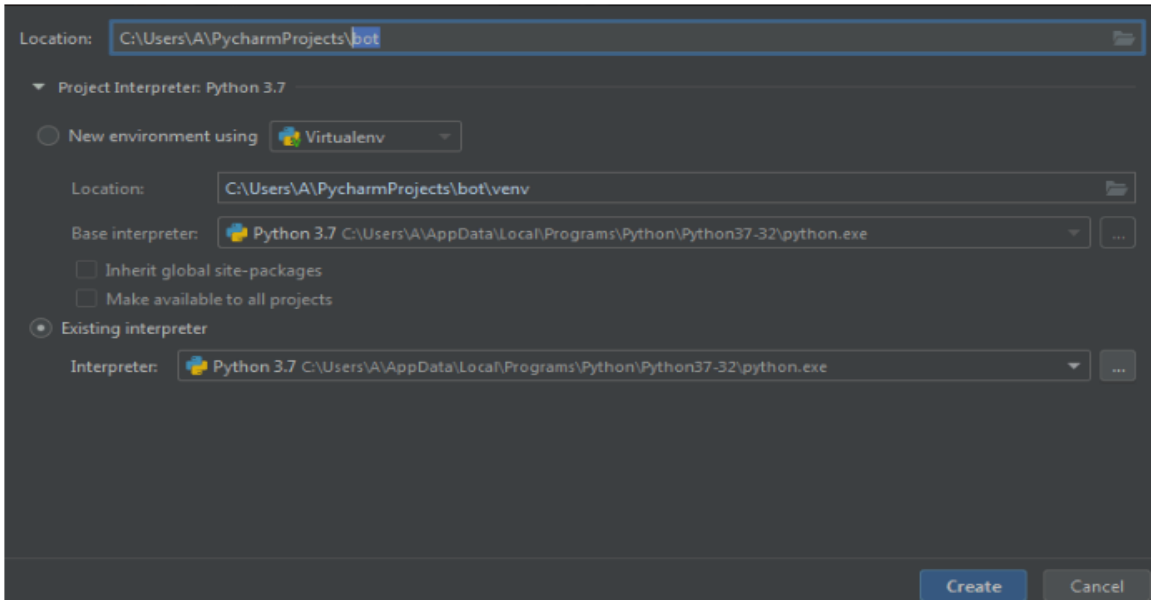
Для начала нужно скачать python на официальном сайте. Не забудьте поставить галочку add to PATH во время установки! После установки python'а нам понадобится хороший редактор кода например PyCharm. Осталось скачать библиотеку telebot. Для этого заходим в командную строку и пишем: `pip install pytelegrambotapi`

В поиске telegram находим Bot Farher'а и создаем своего бота с помощью команды /newbot. Затем вводим имя и юзернейм. Обратите внимание, что юзернейм должен оканчиваться на bot!



Как вы видите нам выдали специальный api токен, с помощью которого вы сможете управлять своим ботом (в моём случае это: *776550937:AAELEr0c3H6dM-9QnlDD-0Q0Fcd65pPyAiM*). Свой токен Вы можете запомнить, но я рекомендую его записать.

Настал момент, которого ждали все. Открываем PyCharm и создаем новый проект.



Рекомендую поставить всё как у меня (название, конечно можно изменить). После создания проекта, давайте создадим файл, в котором будет наш код. Кликните правой кнопкой по папке с вашем проектом, затем *New → Python File*. Теперь напишем сам код. Импортируем библиотеку telebot, с помощью:

```
import telebot
```

Теперь нужно создать переменную bot. На самом деле имя переменной может быть каким угодно, у меня bot.

```
bot = telebot.TeleBot('ваш токен')
```

Напишем декоратор bot.message_handler(), с помощью которого наш бот будет реагировать на команду /start. Для этого в круглых скобках пишем commands=[‘start’]. В итоге у нас должно получиться это:

```
@bot.message_handler(commands=['start'])
```

Если Вы попробуете запустить своего бота (ПКМ->Run), то у вас ничего не выйдет. Во первых в конце кода мы должны прописать bot.polling(). Это нужно для того, чтобы бот не выключился сразу, а работал и проверял, нет ли на сервере нового сообщения. А во вторых наш бот если уж и будет проверять наличие сообщений, то всё равно ничего ответить не сможет. Пора это исправлять! После нашего декоратора создаем функцию start_message, которая будет принимать параметр message (название функции может быть любым). Далее реализуем отправку сообщения от самого бота. В функции пропишем bot.send_message(message.chat.id, ‘Привет, ты написал мне /start’). Смотрите, что у Вас должно получиться:

```
import telebot
bot = telebot.TeleBot('776550937:AAELEr0c3H6dM-9QnlDD-0Q0Fcd65pPyAiM')
@bot.message_handler(commands=['start'])
def start_message(message):
    bot.send_message(message.chat.id, 'Привет, ты написал мне /start')
bot.polling()
```

Должно быть следущее

Чтобы он отвечал не только на команды, но и на сообщения, создадим новый декоратор bot.message_handler(), а в круглые скобочки напишем content_types=[‘text’]. Вообще

существует множество видов контента, к примеру location, photo, audio, sticker и т.д. Но нам же нужно отвечать на текст, верно? Поэтому создаём функцию send_text, принимающую параметр message. В функции пропишем условие:

```
@bot.message_handler(content_types=['text'])
def send_text(message):
    if message.text == 'Привет':
        bot.send_message(message.chat.id, 'Привет, мой создатель')
    elif message.text == 'Пока':
        bot.send_message(message.chat.id, 'Прощай, создат
```

Если текст сообщения будет равен «Привет», то бот отвечает «Привет, мой создатель», а если текст сообщения будет равен «Пока», то бот ответит «Прощай, создатель». Тут думаю всё понятно. Но вы скорее всего задались вопросом, а если пользователь пропишет «привет», ну или «пРиВет», как быть в этой ситуации? Всё достаточно просто! В условии, после message.text напишите функцию .lower(), а в тексте все заглавные буквы замените на строчные. Теперь наш бот отвечает не только на «привет», но и на «ПриВеТ», и даже «пРиВеТ».

Вот что у вас должно получиться:

```
import telebot
bot = telebot.TeleBot('776550937:AAELEr0c3H6dM-9QnlDD-0Q0Fcd65pPyAiM')
@bot.message_handler(commands=['start'])
def start_message(message):
    bot.send_message(message.chat.id, 'Привет, ты написал мне /start')
@bot.message_handler(content_types=['text'])
def send_text(message):
    if message.text.lower() == 'привет':
        bot.send_message(message.chat.id, 'Привет, мой создатель')
    elif message.text.lower() == 'пока':
        bot.send_message(message.chat.id, 'Прощай, создатель')
bot.polling()
```

Как же отправить к примеру стикер. У каждого стикера есть свой id, соответственно зная id мы сможем его отправить. Получить id стикера можно двумя способами. Первый (простой) — через специального бота «What’s the sticker id?»

Ну и второй способ, для тех, кто не ищет лёгких путей. Создаем новый декоратор bot.message_handler(), вот только в скобочки пишем content_types=[‘sticker’]. Далее всё как обычно. Создаем функцию, принимающую параметр message, а вот в ней пропишем print(message). Запускаем бота.

Давайте сделаем так, чтобы когда пользователь отправил боту «я тебя люблю», то бот ему ответил стикером. Создавать новый декоратор не нужно, мы просто допишем условие, которое было до этого. Вот только вместо bot.send_message() пропишем bot.send_sticker(), а вместо текста напишем id стикера.

Расмотрим, как сделать клавиатуру, которую бот покажет вам при старте. Это уже будет сделать сложнее. Создаем переменную keyboard1, в которую запишем telebot.types.ReplyKeyboardMarkup(). Эта функция вызывает клавиатуру. Далее создадим ряды, но помните, что рядов может быть не больше 12! Для того, чтобы их создать, пишем keyboard1.row(). В круглые скобочки запишите всё что хотите, лично я напишу «Привет» и «Пока». Теперь, чтобы вызвать клавиатуру, допишем reply_markup=keyboard1 к функции отправки сообщения при старте. Вот, что у вас должно получиться:

```
keyboard1 = telebot.types.ReplyKeyboardMarkup()
keyboard1.row('Привет', 'Пока')
@bot.message_handler(commands=['start'])
def start_message(message):
    bot.send_message(message.chat.id, 'Привет, ты написал мне /start', reply_markup=keyboard1)
```

Вы видите, что клавиатура какая-то большая. Чтобы это исправить, нужно просто в ReplyKeyboardMarkup() прописать True. Ну а если вы хотите, чтобы клавиатура скрывалась, как только пользователь нажал на нее, то напишите еще один True.

```
keyboard1 = telebot.types.ReplyKeyboardMarkup(True, True)
```

Исходный код:

```
import telebot
bot = telebot.TeleBot('<ваш токен>')
keyboard1 = telebot.types.ReplyKeyboardMarkup()
keyboard1.row('Привет', 'Пока')
@bot.message_handler(commands=['start'])
def start_message(message):
    bot.send_message(message.chat.id, 'Привет, ты написал мне /start', reply_markup=keyboard1)
@bot.message_handler(content_types=['text'])
def send_text(message):
    if message.text.lower() == 'привет':
        bot.send_message(message.chat.id, 'Привет, мой создатель')
    elif message.text.lower() == 'пока':
        bot.send_message(message.chat.id, 'Прощай, создатель')
    elif message.text.lower() == 'я тебя люблю':
        bot.send_sticker(message.chat.id, 'CAADAgADZgkAAnlc4gmfCor5YbYYRAI')
@bot.message_handler(content_types=['sticker'])
def sticker_id(message):
    print(message)
bot.polling()
```

Что такое message и почему, чтобы отправить сообщение мы должны указать message.chat.id в параметрах функции send_message? Для того, чтобы узнать это давайте выведем message в консоль:

```
@bot.message_handler(commands=['start'])
def start(message):
    print(message)
```

Теперь когда мы вводим команду */start*, наш бот присылает огромное кол-во информации. Все, что мы сейчас получили — это ответ в формате json. Json — это простой формат для хранения структурированных данных. Все выводится в формате: ‘ключ’: значение. Давайте посмотрим на то, что получилось


```
{
    'content_type':'text',
    'message_id':573,
    'from_user':{
        'id':687595402,
        'is_bot':False,
        'first_name':'Dmitry',
        'username':'dimagorovtsov',
        'last_name':'Gorovtsov',
        'language_code':'ru'
    },
    'date':1565206363,
    'chat':{
        'type':'private',
        'last_name':'Gorovtsov',
        'first_name':'Dmitry',
        'username':'dimagorovtsov',
        'id':687595402,
        'title':None,
        'all_members_are_administrators':None,
        'photo':None,
        'description':None,
        'invite_link':None,
        'pinned_message':None,
        'sticker_set_name':None,
```

```
            'can_set_sticker_set':None
        },
        'forward_from_chat':None,
        'forward_from':None,
        'forward_date':None,
        'reply_to_message':None,
        'edit_date':None,
        'media_group_id':None,
        'author_signature':None,
        'text':'/start',
        'entities':[
            <telebot.types.MessageEntity object at 0x03807F50>
        ],
        'json':{
            'message_id':573,
            'from':{
                'id':687595402,
                'is_bot':False,
                'first_name':'Dmitry',
                'last_name':'Gorovtsov',
                'username':'dimagorovtsov',
                'language_code':'ru'
            },
            'chat':{
                'id':687595402,
                'first_name':'Dmitry',
                'last_name':'Gorovtsov',
                'username':'dimagorovtsov',
                'type':'private'
            },
            'date':1565206363,
            'text':'/start',
            'entities':[
                {
                    'offset':0,
                    'length':6,
                    'type':'bot_command'
                }
            ]
        }
    }
```


К примеру из всей этой информации мы хотим получить id чата, из которого мы отправли сообщение. Для этого обратимся к ключу chat.

Запрос:

```
print(message.chat)
```

Ответ:


```
{'type': 'private', 'last_name': 'Gorovtsov', 'first_name': 'Dmitry', 'username': 'dimagorovtsov', 'id': 687595402, 'title': None, 'all_members_are_administrators': None, 'photo': None, 'description': None, 'invite_link': None, 'pinned_message': None, 'sticker_set_name': None, 'can_set_sticker_set': None}
```


Смотрите, у ключа chat есть еще несколько ключей: first_name, last_name, username… и у каждого из них есть свои значения. Теперь обратимся к ключу id:

```
print(message.chat.id)
```

Как вы видите для того чтобы получить нужное значение необходимо просто записать название ключей через точку. А теперь смотрим на ответ от сервера:

687595402

Все идет как надо! Мы получили id чата, собственно как и хотели! А теперь получим имя отправителя. Тут, как вы заметили нужно использовать ключ from_user.

Запрос:

print(message.from_user)

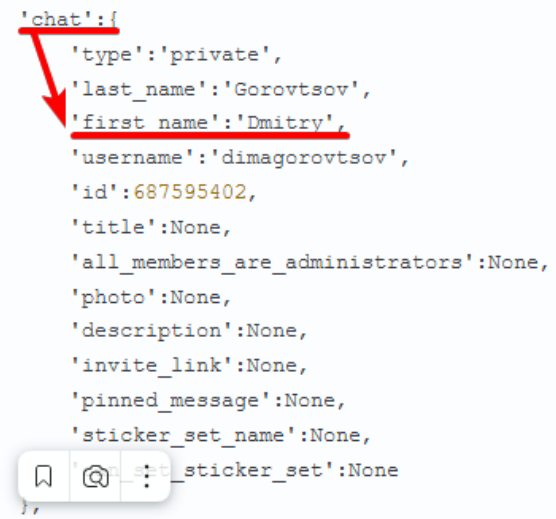
Ответ:

{'id': 687595402, 'is_bot': False, 'first_name': 'Dmitry', 'username': 'dimagorovtsov', 'last_name': 'Gorovtsov', 'language_code': 'ru'}

Теперь достанем значение у ключа first_name:

print(message.from_user.first_name)

Ну вот и все! За пару секунд мы смогли получить id чата и мое имя в telegram. И еще раз, для тех кто не понял:



Чтобы получить значение ключа first_name, нам нужно сначала обратиться к ключу chat, а только потом уже к first_name!

Теперь смотрите, для того, чтобы отправить сообщение в какой-либо чат нам необходимо указать несколько параметров в функцию send_message. Первый параметр — это chat_id, собственно сам id чата. Второй — text, текст сообщения. И как вы догадались, вместо того, чтобы писать message.chat.id, мы можем написать свои данные! Вот так можно прислать сообщение самому себе, если указать в параметрах свой id:

bot.send_message(<ваш id>, 'Текст сообщения')

Ну а когда мы пишем message.chat.id, мы подразумеваем, что бот отправит сообщение в чат, из которого его вызвали.

Лабораторная работа № 3. Технология Интернет вещей.

***Цель работы**: знакомство студентов с основными понятиями языка HTML, структурой HTML-документа, обязательными метками, комментариями, форматирования текста, физическими и логическими стилями, приобретение создания простейших статических Web-документов.*

Задание № 1. ***Создание простейшего файла HTML***

1.Создайте личную папку, куда вы будете сохранять все файлы своего сайта.

2.Запустите программу *Блокнот* (*Notepad*).

3.Наберите в окне программы простейший файл HTML.

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
Расписание занятий на вторник
</BODY>
```

</HTML>

4. Сохраните файл под именем **RASP.HTML** (обязательно укажите тип файла HTML при сохранении) в личной папке.

5. Для просмотра Web-страницы используйте любую программу браузера (*Internet Explorer, Opera, Mozilla Firefox* или другую). Для этого, не покидая программу Блокнот (сверните окно на панель задач), откройте личную папку и двойным кликом по файлу RASP.HTML откройте окно браузера.

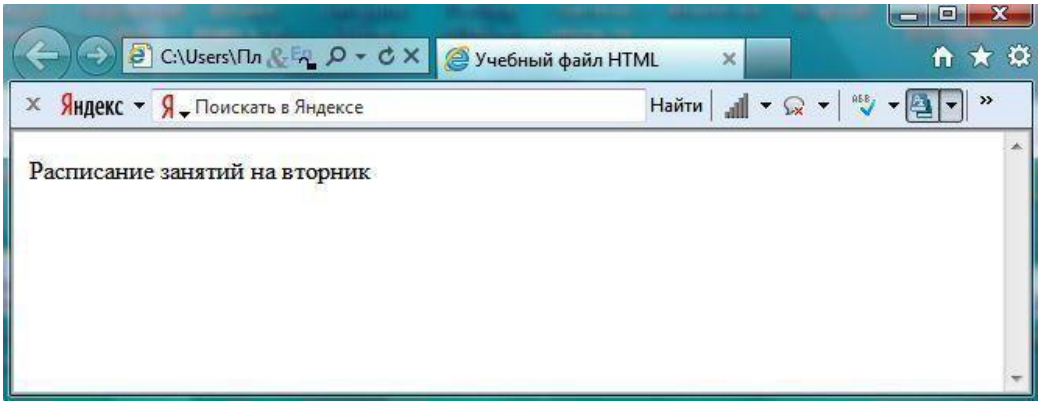


Рис.1

На экране вы увидите результат работы, изображенный на рисунке 1.

Задание № 2. ***Управление расположением текста на экране***

1. При необходимости откройте текст Web-страницы в Блокноте (1 щелчок правой клавишей мыши по файлу RASP.HTML, в контекстном меню выбрать команду *Открыть с помощью...* и выбрать программу *Блокнот*). При необходимости открыть файл в браузере – двойной клик по значку файла левой клавишей мыши.
2. Внести изменения в файл RASP.HTML, расположив слова *Расписание, занятий, на вторник* на разных строках.
3. <HTML>
   <HEAD>
   <TITLE>Учебный файл HTML</TITLE>
   </HEAD>
   <BODY>
   Расписание занятий
   на вторник
   </BODY>
   </HTML>

4. Сохраните текст с внесенными изменениями в файле RASP.HTML (меню Файл | Сохранить). Если у вас уже отображается Web-страница, то вам достаточно переключиться на панели задач на программу браузера и обновить эту страницу (кнопка ↻). Изменилось ли отображение текста на экране?Не удивляйтесь тому, что внешний вид вашей Web-страницы не изменился.

Не забывайте каждый раз сохранять текст Web-страницы при ее корректировке в программе *Блокнот* и обновлять страницу при ее просмотре в программе браузера.

Задание № 3. ***Некоторые специальные команды форматирования текст***

Существуют специальные команды, выполняющие перевод строки и задающие начало нового абзаца. Кроме того существует команда, запрещающая программе браузера изменять каким-либо образом изменять форматирование текста и позволяет точно воспроизвести на экране заданный фрагмент текстового файла.

Тег перевода строки **
** отделяет строку от последующего текста или графики.

Тег абзаца **<P>** тоже отделяет строку, но еще добавляет пустую строку, которая зрительно выделяет абзац.

Оба тега являются одноэлементными, тег <P> – двойной, т.е. требуется закрывающий тег.

1. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
<P>Расписание</P>
<BR>занятий<BR> на вторник
</BODY>
</HTML>
```

2. Сохраните внесенные изменения, переключитесь на панели задач на программу браузера, обновите Web-страницу.

Как изменилось отображение текста на экране? Выглядеть ваша Web-страница будет примерно так, как показано на рисунке 2.

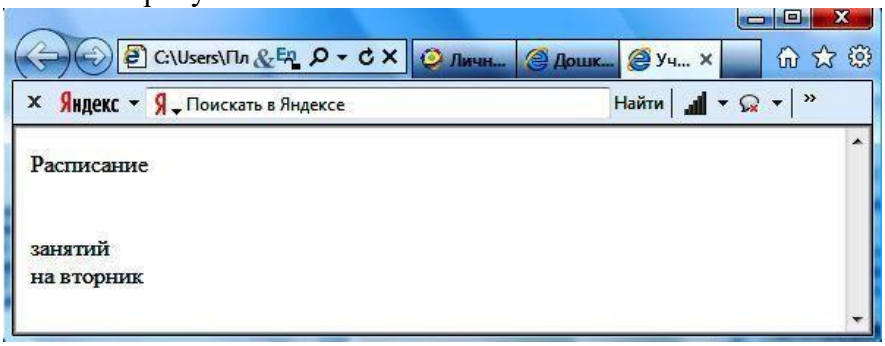


Рис. 2

Задание № 4. ***Выделение фрагментов текста***

1. Внести изменения в текст файла RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
<B>Расписание</B>
<I> занятий</I>
<U> на вторник</U>
</BODY>
</HTML>
```

2. Посмотрите полученную Web-страницу.

Возможно использование комбинированных выделений текста.

<I><B>Расписание</B></I> <I><U> занятий</U></I> <U> на вторник</U>

Но при этом необходимо помнить следующее правило использования комбинированных тегов:

<Тег_1><Тег_2> ... </Тег_2></Тег_1> – ***правильная запись***.

<Тег_1><Тег_2> ... </Тег_1></Тег_2> – ошибочная запись.

Обратите внимание на «вложенность» тегов, она напоминает «вложенность» скобок.

Задание № 5. ***Задание размеров символов Web-страницы***

Существует два способа управления размером текста, отображаемого браузером:

- использование стилей заголовка,
- задание размера шрифта основного документа или размера текущего шрифта.

Используется шесть тегов заголовков: от **<H1>** до **<H6>** (тег двойной, т.е. требует закрытия). Каждому тегу соответствует конкретный стиль, заданный параметрами настройки браузера.

1. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
```

```
<BODY>
<P><H1>Расписание</H1></P>
<I> занятий</I><U> на вторник</U>
</BODY>
</HTML>
```

2. Просмотрите свою Web-страницу. На экране вы увидите то, что отображено на рисунке 3

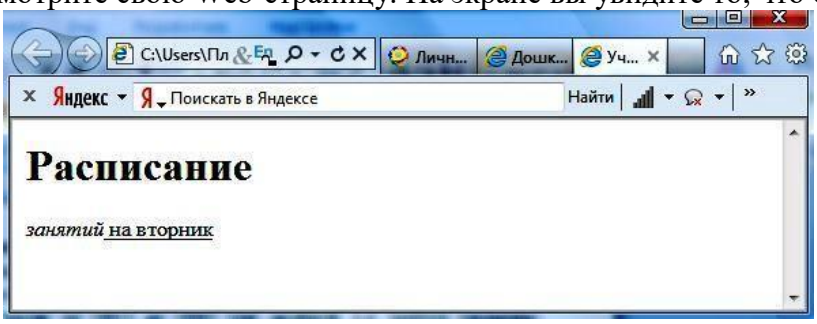


Рис. 3

Задание № 6. ***Установка размера текущего шрифта***

Тег шрифта **<FONT>** позволяет задавать размер текущего шрифта в отдельных местах текста в диапазоне от 1 до 7.

1.Внесите изменения в текст RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
<FONT SIZE="7">Расписание</FONT> занятий на вторник
</BODY>
</HTML>
```

2.Самостоятельно измените размер текста «занятий на вторник», используя тег <FONT>.

3.Измените оформление текста HTML-документа, используя тег выделения фрагментов и тег перевода строки и абзаца.

Задание № 7. ***Установка гарнитуры и цвета шрифта***

Тег **<FONT>** предоставляет возможности управления гарнитурой, цветом и размером текста. Изменение гарнитуры текста выполняется простым добавлением к тегу **<FONT>** атрибута **FACE**. Например, для отображения текста шрифтом *Arial* необходимо записать:

<FONT FACE=”ARIAL”>

Для изменения цвета шрифта можно использовать в теге **<FONT>** атрибут **COLOR=”X”**. Вместо **“X”** надо подставить английское название цвета в кавычках (“ ”), либо его шестнадцатеричное значение. При задании цвета шестнадцатеричным числом необходимо представить этот цвет разложенным на три составляющие: красную (*R – Red*), зелѐную (*G – Green*), синюю (*B – blue*), каждая из которых имеет значение от **00** до **FF**. В этом случае мы имеем дело с так называемым форматом **RGB**. Примеры записи текста в формате **RGB** приведены в Таблице 1:

Таблица 1Запись текста в формате RGB

Цвет	RRGGBB	Цвет	RRGGBB	Цвет	RRGGBB
Black Черный	000000	Purple Фиолетовый	FF00FF	Green Зеленый	00FF00
White Белый	FFFFFF	Yellow Желтый	FFFF00	Azure Бирюзовый	00FFFF
Red Красный	FF0000	Brown Коричневый	996633	Blue Синий	0000FF
Orange Оранжевый	FF8000	Violet Лиловый	B000FF	Gray Серый	A0A0A0

1. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
<U><I><B><FONT COLOR="#FF0000" FACE="ARIAL" SIZE="7">
Расписание</FONT></B></I></U> занятий на вторник
</BODY>
</HTML>
```

2. Самостоятельно измените размер, цвет, гарнитуру стиль текста документа.

Задание № 8. ***Выравнивание текста по горизонтали***

1. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY>
<P ALIGN="CENTER">
<FONT COLOR="#008080" SIZE="7">
<B>Расписание</B></FONT><BR>
<FONT SIZE="6"><I> занятий на вторник</I></FONT>
</P>
</BODY>
</HTML>
```

2. Просмотрите изменения в браузере. На экране вы увидите то, что показано на рисунке 4.

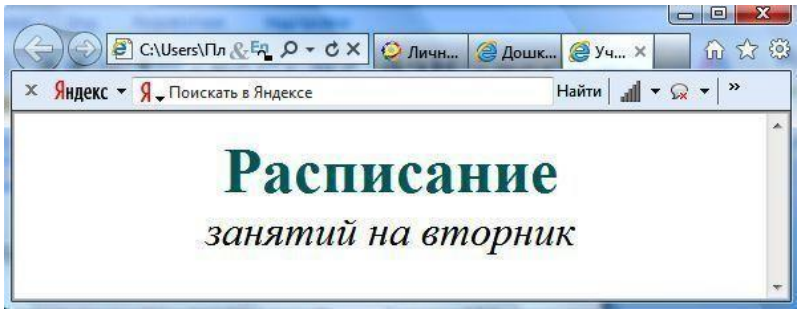


Рис. 4

Задание № 9. ***Задание цвета фона и текста***

При изображении фона и цвета браузеры используют цвета, установленные по умолчанию, – они заданы параметрами настройки браузера. Если вы хотите задать другие цвета, то это надо сделать в начале файла HTML в теге **<BODY>**. Атрибут **BGCOLOR=** определяет цвет фона страницы, атрибут **TEXT=** задает цвет текста для всей страницы, атрибуты **LINK=** и **VLINK=** определяют соответственно цвета непросмотренных и просмотренных ссылок (последние два примера будут рассмотрены позже).

1. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFCC" TEXT="#330066">
<P ALIGN="CENTER">
<FONT COLOR="#008080" SIZE="7">
<B>Расписание</B></FONT><BR>
<FONT SIZE="6"><I> занятий на вторник</I></FONT>
</P>
```

```
</BODY>
</HTML>
```

2. Просмотрите изменения Web-страницы в браузере.

Задание № 10. ***Размещение графики на Web-странице***

Тег **<IMG>** позволяет вставить изображение на Web-страницу. Оно появится в том месте документа, где находится этот тег. Тег **<IMG>** является одиночным.

Необходимо помнить, что графические файлы должны находиться в той же папке, что и файл HTML, описывающий страницу. Графика в Web, как правило, распространяется в трех форматах: GIF, JPG, PNG.

Для выполнения следующего задания поместите файл с именем CLOCK.JPG (или другим именем) в рабочую папку.

Следует помнить, что для браузера важно, в каком регистре вы задаете описание имени и типа файла. Выработайте для себя определенное правило и строго следуйте ему. Если вы размещаете файл графического изображения во вложенной папке, то при описании изображения необходимо указывать путь доступа к файлу изображения, отображая вложенность папок.

1. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFCC" TEXT="#330066">
<P ALIGN="CENTER">
<FONT COLOR="#008080" SIZE="7">
<B>Расписание</B></FONT><BR>
<FONT SIZE="6"><I> занятий на вторник</I></FONT>
<BR><BR>
<IMG SRC="CLOCK.PNG">
</P>
</BODY>
</HTML>
```

2. Просмотрите изменения вашей Web-страницы в браузере.

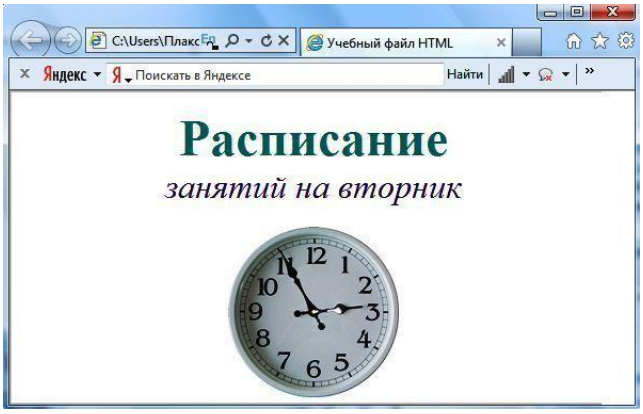


Рис. 5

Тег <IMG> имеет немало атрибутов, описанных в таблице 2. Эти атрибуты можно задавать дополнительно и располагаться они могут в любом месте тега после кода **IMG**.

Таблица 2 **Атрибуты изображения**

Атрибут	Формат	Описание
ALT	<IMG SRC="CLOCK.JPG" ALT="Картинка">	Задает текст, заменяющий изображение в том случае, если браузер не воспринимает изображение
BORDER	<IMG SRC="CLOCK.JPG" BORDER="3">	Задает толщину рамки вокруг изображения. Измеряется в пикселях

ALIGN	<IMG SRC="CLOCK.JPG" ALIGN="TOP">	Задает выравнивание изображения относительно текста: • относительно текста выровнена верхняя часть изображения – **"TOP"**, • относительно текста выровнена нижняя часть изображения – **"BOTTOM"**, • относительно текста выровнена средняя часть изображения – **"MIDDLE"**.
HEIGHT	<IMG SRC="CLOCK.JPG" HEIGHT="111">	Задает вертикальный размер изображения внутри окна браузера
WIDTH	<IMG SRC="CLOCK.JPG" WIDTH="210">	Задает горизонтальный размер изображения внутри окна браузера
VSPACE	<IMG SRC="CLOCK.JPG" VSPACE="8">	Задает добавление верхнего и нижнего пустых полей
HSPACE	<IMG SRC="CLOCK.JPG" HSPACE="8">	Задает добавление левого и правого пустых полей

Задание № 11. ***Использование атрибутов изображения***

1. Самостоятельно внесите изменения в текст файла RASP.HTML: опробуйте использование таких атрибутов графики, как ALT, BORDER, ALIGN, HEIGHT, WIDTH, VSPACE, HSPACE.

Всегда обращайте внимание на размер графического файла (в байтах), так как это влияет на время загрузки Web-страницы.

2. Просмотрите изменения вашей Web-страницы в браузере.

Задание № 12. ***Установка фонового изображения на Web-странице***

Фоновое изображение – это графический файл с небольшим рисунком, который многократно повторяется, заполняя все окно браузера независимо от его размеров.

Графика, используемая в качестве фоновой, задается в теге **<BODY>**.

1. Внесите изменения в файл RASP.HTML, предварительно подготовив и сохранив в рабочей папке графический файл фонового рисунка (FON.PNG).

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY BACKGROUND="FON.PNG" TEXT="#330066">
<P ALIGN="CENTER">
<FONT COLOR="#008080" SIZE="7">
<B>Расписание</B></FONT><BR>
<FONT SIZE="6"><I> занятий на вторник</I></FONT>
<BR><BR>
<IMG SRC="CLOCK.PNG" ALIGN="MIDDLE">
</P>
</BODY>
</HTML>
```

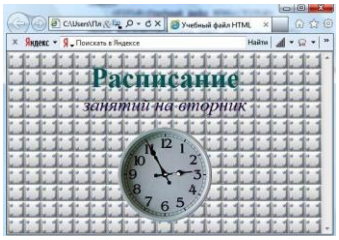


Рис. 6

Рисунок, который использовался в качестве фонового, имеет вид

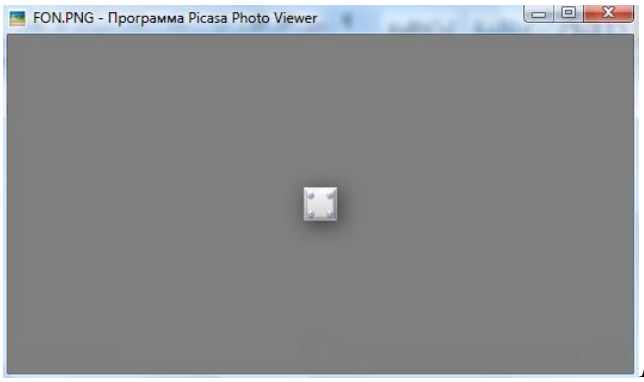


Рис. 7

2. Поэкспериментируйте с фоновым рисунком Web-страницы и выберите оптимальный с вашей точки зрения.

Задание № 13. ***Создание таблицы***

Таблица является частью HTML-документа. Она представляет собой прямоугольную сетку, состоящую из вертикальных столбцов и горизонтальных строк. Пересечение строки и столбца называется ячейкой таблицы. Ячейка может содержать в себе текст, графику или другую таблицу.

Таблица состоит из трех основных частей:

- названия таблицы,
- заголовков столбцов,
- ячеек таблицы.

Таблица в Web-документе заполняется по строкам (слева направо по строке, затем переход на новую строку). Каждая ячейка таблицы должна быть заполнена (хотя бы пробелом, которые используются для создания пустых ячеек).

1. Запустите программу *Блокнот* и наберите текст следующей Web-страницы. Применяйте приемы копирования при создании таблицы, работая в программе *Блокнот*.

```
<HTML>
<HEAD>
<TITLE>Расписание занятий 5 классов</TITLE>
</HEAD>
<BODY BGCOLOR="FFFFFF">
<P ALIGN="CENTER">
<FONT COLOR="RED" SIZE="6" FACE="ARIAL">
<B>5 класс</B></FONT><BR></P>
<FONT COLOR="BLUE" SIZE="4" FACE="COURIER">
<B>Понедельник</B></FONT><BR>
<TABLE BORDER="1" WIDTH=100% BGCOLOR="#99CCCC">
<TR BGCOLOR="#CCCCFF" ALIGN="CENTER">
<TD>Урок</TD> <TD>5 "А"</TD> <TD>5 "Б"</TD>
<TD>5 "В"</TD>
</TR>
<TD>1</TD> <TD>Русский язык</TD>
<TD>Литература</TD>        <TD>История</TD>
</TR>
<TD>2</TD> <TD>Математика</TD>
<TD>Информатика</TD> <TD>Английский язык</TD>
</TR>
<TD>3</TD> <TD>История</TD>
<TD>Математика</TD> <TD>Информатика</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

2. Сохраните файл в личной рабочей папке под именем 5.HTML

3. Для просмотра созданной Web-страницы в окне личной рабочей папки двойным щелчком левой клавиши мыши загрузите браузер.

Рис. 8

Задание № 14. ***Построение гипертекстовых связей***

Важнейшим средством языка HTML является возможность включения в документ ссылок на другие документы.

Возможны ссылки:

- на удаленный HTML-файл,
- на некоторую точку в текущем HTML-документе,
- на любой файл, не являющийся HTML-документом.

В качестве ссылки можно использовать любой текст или графику.

Ссылки в пределах одного документа

Такие ссылки требуют двух частей: метки и самой ссылки. Метка определяет точку, на которую происходит переход по ссылке. Ссылка использует имя метки. Ссылки выделяют цветом или подчеркиванием в зависимости от того, как настроен браузер. Для изменения цвета ссылки используются атрибуты **LINK=** и **VLINK=** тега **<BODY>**.

Описание ссылки

<A HREF="#ПН">Понедельник</A>

Перед именем метки (ПН), указывающей, куда надо перейти по ссылке, ставится символ #. Между символами “>” и “<” располагается текст (“Понедельник”), на котором должен быть произведен щелчок для перехода по ссылке.

Определим метку

<A NAME="ПН">Понедельник</A>

1.Дополните файл 5.HTML описанием таблицы, содержащей названия дней недели, поместив его в начало Web-страницы.

```
<TABLE WIDTH=100%>
<TR>
<TD>Понедельник</TD>
<TD>Вторник</TD>
<TD>Среда</TD>
<TD>Четверг</TD>
<TD>Пятница</TD>
<TD>Суббота</TD>
</TR>
</TABLE>  <BR>
```

2.Вставьте в файл 5.HTML метку, указывающую на понедельник.

```
<FONT COLOR="BLUE" SIZE="4" FACE="COURIER"><B>
<A NAME="ПН">Понедельник</A></B></FONT><BR>
```

3.Вставьте в таблицу с названиями дней недели ссылку для выбранной метки:

```
<TABLE WIDTH=100%>
<TR>
<TD><A HREF="#ПН">Понедельник</A></TD>
<TD>Вторник</TD>
<TD>Среда</TD
```

4.Создайте таблицы расписаний для других дней недели.
5.Сохраните файл 5.HTML в личной рабочей папке.
6.Просмотрите полученную Web-страницу.

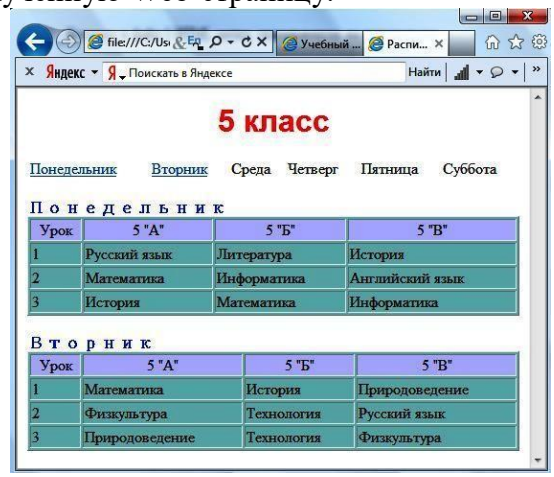


Рис. 9

Задание № 15. ***Создание ссылки на другой HTML-документ***

Ссылки позволяют щелчком на выделенном слове или фразе перейти к другому файлу. Опишем ссылку:

<A HREF="5.HTML">5 класс</A>

После имени файла (5.HTML) между символами «>» и «<» располагается текст («5 класс»), на котором должен быть произведен щелчок для перехода к этому файлу.

1. Внесите изменения в файл RASP.HTML

```
<HTML>
<HEAD>
<TITLE>Учебный файл HTML</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF" TEXT="#330066">
<P ALIGN="CENTER">
<FONT COLOR="#008080" SIZE="7">
<B>Расписание</B></FONT><BR>
<FONT SIZE="6"><I> занятий на вторник</I></FONT>
<BR><BR>
<IMG SRC="CLOCK.PNG" ALIGN="TOP">
</P>
<CENTER>
<TABLE WIDTH=60%>
<TR><TD><A HREF="5.HTML">5 класс</A></TD>
<TD>6 класс</TD>     </TR>
<TR><TD>7 класс</TD>
<TD>8 класс</TD>     </TR>
<TR><TD>9 класс</TD>
<TD>10 класс</TD>   </TR>
<TR><TD>11 класс</TD>     </TR>
</TABLE>
</CENTER>
</BODY>
</HTML>
```

2. Сохраните файл RASP.HTML
3. Просмотрите полученную Web-страницу.

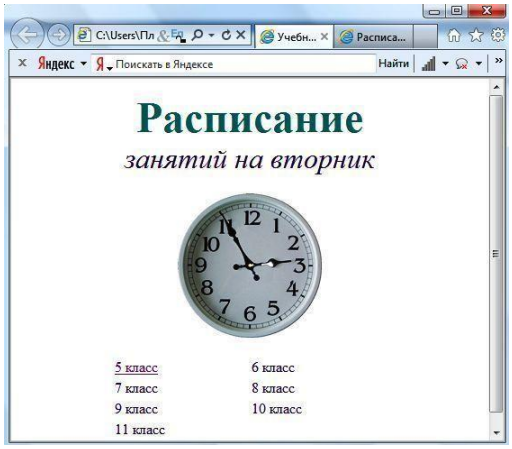


Рис. 10

Задание № 16. ***Создание ссылки на другой HTML-документ***

1. Внесите изменения в файл 5.HTML так, чтобы в конце страницы была ссылка на головную страницу Расписание занятий 5 классов (RASP.HTML). В качестве ссылки используйте графический файл (HOME.GIF) следующим образом:

```
</TABLE><BR>
<CENTER>
<A HREF="RASP.HTML"><IMG SRC="HOME.PNG" BORDER="0"></A>
</CENTER>
```

2.Просмотрите полученную Web- страницу. На экране вы увидите то, что показано на рисунке 11. В качестве ссылки выступает рисунок – стрелка.

Рис. 11

Задание № 17. ***Самостоятельное итоговое задание***

Разработайте Web-страницы, рассказывающие о вашем классе. На головной странице разместите рассказ о классе, классном руководителе. Рассказы об учениках разместите на отдельных Web- страницах. Укажите ссылки на страницы учеников с головной Web-страницы. Не забудьте разместить ссылки возврата на головную страницу.

Как подготовить хорошую Web-страницу

1. Следует обратить внимание на простоту и логичность расположения информации на ваших страницах. Один из способов сделать информацию более легкой для восприятия – оставить на странице достаточно свободного места, не содержащего ни текста, ни рисунков. Страница, содержащая много информации, только отпугнет посетителя.
2. Постарайтесь представить информацию в виде списков или таблиц так, чтобы можно было достаточно легко найти важные сведения.
3. Не размещайте одно изображение сразу за другим. Попробуйте распределить их по документу, оставив достаточно свободного пространства.
4. Информация должна размещаться частями, легкими для восприятия. Обратите внимание на длину абзацев. Если абзац слишком длинный, разбейте его на несколько небольших

абзацев.

5. Если Web-страница имеет большой объем, то, возможно, вам следует вставить ссылки, позволяющие пользователю быстро перемещаться между частями одного документа. Иногда имеет смысл вместо одного длинного документа подготовить одну страницу, содержащую перечень тем, каждую из которых раскрыть на отдельной Web-странице, и установить ссылки на соответствующие Web-страницы.

6. Использование графики может дополнительно привлечь пользователей. Но необходимо помнить о времени загрузки вашей страницы, которое определяется количеством и объемом графической информации. Красивая картинка не произведет никакого впечатления, ели для того, чтобы ее увидеть, придется долго ждать, пока она загрузится.

Тестирование

Перед тем как выставлять свои Web-страницы на сервер необходимо их протестировать. Созданные документы должны пройти «локальную проверку» в пределах вашего жесткого диска. При проверке используйте разные браузеры. Вы увидите различия, которые могут оказаться довольно существенными.

В рамках тестирования необходимо сделать следующее:

1. *Проверить правописание.* Выполните автоматизированную проверку правописания текста (для этого можно использовать Microsoft Word) или попросите кого-нибудь выполнить корректуру.

2. *Проверит навигацию.* Убедитесь, что на каждой странице присутствуют необходимые средства навигации, все ссылки работают правильно.

3. *Проверить доступ к внешним файлам.* Выясните, размещены ли графические, звуковые или видеофайлы там, где они могут быть найдены и откуда их можно загрузить (должен быть правильно указан путь доступа). Для неграфических браузеров требуется задать подменяющие текстовые сообщения.

4. *Проверить, допустимо ли время загрузки.*

5. *Осуществить проверку ваших Web-страниц посторонним лицом.* Попросите кого-нибудь, кто не знаком с вашими документами, пройти их от начала до конца. Иногда при этом выясняются такие факты, каких вы сами ни за что бы не заметили.

Индивидуальные задания к лабораторной работе №3.

1. Магазин техники
2. Аптека.
3. Магазин музыкальных инструментов.
4. Магазин одежды.
5. Магазин посуды.
6. Автосалон.
7. Магазин стройматериалов.
8. Магазин оптики.
9. Магазин мебели
10. Магазин игрушек
11. Магазин канцелярских товаров.
12. Магазин зоотоваров.
13. Магазин алкогольной продукции.
14. Исламский магазин
15. Сувенирная лавка
16. Магазин бытовой химии.

Лабораторная работа № 4. Введение в машинное обучение.

Цель работы: *анализ программы для распознавания образов.*

Каждый раз, когда вы загружаете фотографию в Facebook, платформа использует алгоритмы распознавания лиц для идентификации людей на этом изображении? Или что

определенные правительства во всем мире используют технологию распознавания лиц для выявления и поимки преступников?

Чтобы понять, как работает распознавание лиц, сначала разберемся с понятием вектора признаков. Каждый алгоритм машинного обучения принимает набор данных в качестве входных данных и учится на этих данных. Алгоритм просматривает данные и выявляет закономерности. Например, предположим, что мы хотим определить, чье лицо присутствует на данном изображении, есть несколько вещей, которые мы можем рассматривать как паттерн:

- Высота/ширина лица.
- Высота и ширина могут быть ненадежными, поскольку изображение может быть масштабировано по разному. Однако даже после изменения масштаба неизменными остаются отношения - отношение высоты лица к ширине лица не изменится.
- Цвет лица.
- Ширина других частей лица, таких как губы, нос и т. д.

Понятно, что здесь есть закономерность - разные лица имеют разные размеры, как показано выше. Подобные лица имеют похожие размеры. Задача состоит в том, чтобы преобразовать определенное лицо в числа - алгоритмы машинного обучения понимают только числа. Это числовое представление «лица» (или элемента в обучающем наборе) называется вектором признаков. Вектор признаков состоит из различных чисел в определенном порядке.

В качестве простого примера мы можем отобразить «лицо» в вектор объектов, который может включать в себя различные признаки, такие как: высота лица (см), ширина лица (см), средний цвет лица (R, G, B), ширина губ (см), высота носа (см).

По сути, учитывая изображение, мы можем отобразить различные объекты и преобразовать их в вектор объектов, например:

Высота лица (см) - 23.1 Ширина лица (см) - 15.8

Средний цвет лица (RGB) - (255, 224, 189) Ширина губ (см) - 5.2

Высота носа (см) - 4.4

Итак, наше изображение теперь представляет собой вектор, который можно представить в виде (23.1, 15.8, 255, 224, 189, 5.2, 4.4). Конечно, на изображении может быть множество других черт (например, цвет волос, волосы на лице, очки и т. l.). Однако, для примера, давайте рассмотрим только эти 5 простых признаков.

Теперь, когда мы закодировали каждое изображение в вектор признаков, проблема стала намного проще. Ясно, что когда у нас есть 2 лица (изображения), которые представляют одного и того же человека, полученные векторы признаков будут очень похожи. Иными словами, «расстояние» между двумя векторами признаков будет довольно маленьким.

Машинное обучение может помочь нам с двумя вещами:

1. Получение вектора объекта: трудно вручную перечислить все признаки, потому что их очень много. Алгоритм машинного обучения может разумно обозначить многие из таких признаков. Например, сложными характеристиками могут быть: соотношение высоты носа и ширины лба. Человеку будет довольно сложно перечислить все такие функции «второго порядка».
2. Алгоритмы согласования: после того, как векторы признаков получены, алгоритм машинного обучения должен сопоставить новое изображение с набором векторов признаков, присутствующих в выборке.

Теперь, создадим наш собственный алгоритм распознавания лиц, используя некоторые из известных библиотек Python.

Нам дают несколько лиц - возможно, таких знаменитостей, как Марк Цукерберг, Уоррен Баффет, Билл Гейтс, Шахрукх Кхан и т. д. Назовите эту группу лиц нашей совокупностью. Теперь нам дают изображение еще одной знаменитости («новая знаменитость»). Задача проста - определить, входит ли эта «новая знаменитость» в число присутствующих в совокупности.

Вот некоторые из изображений в совокупности:

Рис.1. Рис.2

Как видите, у нас есть такие знаменитости, как Барак Обама, Билл Гейтс, Джефф Безос, Марк Цукерберг, Рэй Далио и Шахрукх Кхан.

Очевидно, что это Шахрукх Кхан. Однако для компьютера это сложная задача. Проблема заключается в том, что для нас, людей, легко объединить так много признаков изображений, чтобы увидеть, какое из них является знаменитостью. Однако для компьютера непросто научиться распознавать эти лица.

Существует удивительно простая библиотека Python, которая инкапсулирует все, что мы рассматривали выше - создание векторов признаков из лиц и умение различать лица. Эта библиотека Python называется face_recognition и в глубине использует dlib - современный фреймворк C ++, который содержит несколько алгоритмов машинного обучения, которые помогают в написании сложных приложений на C ++.

Библиотека face_recognition в Python может выполнять большое количество задач:

- найти все лица на заданном изображении;
- найти и управлять чертами лица в изображении;
- идентифицировать лица на изображениях;
- распознавать лица в реальном времени.

Здесь мы поговорим о 3-м случае использования - идентификации лиц на изображениях.

Вы можете найти исходный код библиотеки face_recognition здесь, на Github: https://github.com/ageitgey/face_recognition

На самом деле, есть также руководство по установке библиотеки face_recognition: https://github.com/ageitgey/face_recognition#installation-options.

Перед установкой face_recognition вам также необходимо установить dlib. Вы можете найти инструкции по установке dlib здесь:

https://gist.github.com/ageitgey/629d75c1baac34dfa5ca2a1928a7aeaf.

Реализация в Python

Этот раздел содержит код для построения простой системы распознавания лиц с использованием библиотеки face_recognition. Это часть реализации, мы пройдемся по коду, чтобы разобрать его более подробно, в следующем разделе.

```
# импорт библиотек
import os
import face_recognition
# составляем список всех доступных изображений
images = os.listdir('images')
# загружаем ваше изображение
image_to_be_matched = face_recognition.load_image_file('my_image.jpg')
# преобразуем загруженное изображение в вектор признаков
image_to_be_matched_encoded = face_recognition.face_encodings(
    image_to_be_matched)[0]
# итерация для каждого изображения
for image in images:
```

```
    # загружаем изображение
    current_image = face_recognition.load_image_file("images/" + image)
    # преобразуем загруженное изображение в вектор признаков
    current_image_encoded = face_recognition.face_encodings(current_image)[0]
    # проверяем соответствие изображений
    result = face_recognition.compare_faces(
        [image_to_be_matched_encoded], current_image_encoded)
    # проверка совпадения
    if result[0] == True:
        print "Matched: " + image
    else:
        print "Not matched: " + image
```

Структура папок выглядит следующим образом:

```
facialrecognition:
    fr.py
    my_image.jpg
    images/
        barack_obama.jpg
        bill_gates.jpg
        jeff_bezos.jpg
        mark_zuckerberg.jpg
        ray_dalio.jpg
        shah_rukh_khan.jpg
        warren_buffett.jpg
```

Наш корневой каталог, facialrecognition, содержит:

- Наш код распознавания лиц, приведенный выше, в виде fr.py.
- my_image.jpg - изображение, которое нужно распознать («новая знаменитость»).
- images/ - «совокупность».

Когда вы создаете структуру папок, как указано выше, и запускаете приведенный выше код, вот что вы получаете в качестве вывода:

```
Matched: shah_rukh_khan.jpg        Not matched: warren_buffett.jpg
Not matched: barack_obama.jpg      Not matched: ray_dalio.jpg
Not matched: bill_gates.jpg        Not matched: jeff_bezos.jpg
Not matched: mark_zuckerberg.jpg
```

Очевидно, что «новой знаменитостью» является Шахрукх Кхан, и наша система распознавания лиц способна это обнаружить!

Разбор кода Python. Теперь давайте рассмотрим код, чтобы понять, как он работает:

```
# импорт библиотек
import os
import face_recognition
```

Это просто импорт. Мы будем использовать встроенную библиотеку os для чтения всех изображений в нашей совокупности, и будем использовать face_recognition для написания алгоритма.

```
# составляем список всех доступных изображений
images = os.listdir('images')
```

Этот простой код помогает нам определить путь всех изображений в совокупности. Как только эта строка будет выполнена, у нас будет:

```
images = ['shah_rukh_khan.jpg', 'warren_buffett.jpg', 'barack_obama.jpg', 'ray_dalio.jpg',
'bill_gates.jpg', 'jeff_bezos.jpg', 'mark_zuckerberg.jpg']
```

Теперь код ниже загружает изображение новой знаменитости:

```
# загружаем ваше изображение
image_to_be_matched = face_recognition.load_image_file('my_image.jpg')
```

Чтобы убедиться, что алгоритмы способны интерпретировать изображение, мы конвертируем изображение в вектор признаков:

```
# преобразуем загруженное изображение в вектор признаков
image_to_be_matched_encoded = face_recognition.face_encodings(
    image_to_be_matched)[0]
```

Оставшаяся часть кода теперь довольно проста:

```
# итерация для каждого изображения
for image in images:
    # загружаем изображение
    current_image = face_recognition.load_image_file("images/" + image)
    # преобразуем загруженное изображение в вектор признаков
    current_image_encoded = face_recognition.face_encodings(current_image)[0]
    # проверяем соответствие изображений
    result = face_recognition.compare_faces(
        [image_to_be_matched_encoded], current_image_encoded)
    # проверка совпадения
    if result[0] == True:
        print "Matched: " + image
    else:
          print "Not matched: " + image
```

Здесь у нас: итерация цикла на каждом изображении; кодирование изображения в вектор признаков; сравнение загруженного изображения с изображением, которое нужно распознать; если это совпадение, мы печатаем это; если это несоответствие, мы также печатаем это.

Давайте попробуем заменить my_image другим изображением:

При повторном запуске алгоритма вы увидите следующий вывод:

```
Not matched: shah_rukh_khan.jpg  Not matched: warren_buffett.jpg
Not matched: barack_obama.jpg    Not matched: ray_dalio.jpg
Not matched: bill_gates.jpg      Not matched: jeff_bezos.jpg
Not matched: mark_zuckerberg.jpg
```

Очевидно, что система не идентифицировала Джека Ма как кого-либо из вышеупомянутых знаменитостей. Это указывает на то, что наш алгоритм достаточно хорош в обоих случаях.

Задание

Сфотографировать однокурсников и сделать базу как в примере, затем сделать еще фотографию и определить кому она соответствует.

Список литературы.

1. Современные компьютерные технологии : учебное пособие / Р.Г. Хисматов [и др.].. — Казань Казанский национальный исследовательский технологический университет, 2014. — 83 с. — ISBN 978-5-7882-1559-4. — Текст : электронный // IPR SMART : [сайт].
2. Как написать telegram-бота на python с помощью библиотеки telebothttps://bookflow.ru/kak-napisat-telegram-bota-na-python-s-pomoshhyu-biblioteki-telebot/
3. Нейронные сети и генетический алгоритм : учебное пособие / Г. П. Шматов. — Тверь : ТвГТУ, 2019. — 200 с. — ISBN 978-5-7995-1007-7. — Текст : электронный // Лань : электронно-

библиотечная система.