

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания

по выполнению практических работ по дисциплине
«Введение в профессию»
для студентов направления
09.03.02 «Информационные системы и технологии»
шифр и наименование направления подготовки/ специальности

(ЭЛЕКТРОННЫЙ ДОКУМЕНТ)

Невинномысск 2021

ОГЛАВЛЕНИЕ

Введение

1. Практическая работа № 1. Применение облачных сервисов в организации сетевого взаимодействия
2. Практическая работа № 2. Использование Документов Google для организации сетевого взаимодействия
3. Практическая работа № 3. Создание анкет с помощью Документов Google
4. Практическая работа № 4. Создание анкеты «Акклиматизация студентов первокурсников в СКФУ»
5. Практическая работа № 5. Оценка качества Интернет ресурса
6. Практическая работа № 6. Создание личной страницы студента с помощью сервисов Google
7. Практическая работа № 7. Создание личной страницы студента с помощью сервисов Google
8. Практическая работа № 8. Язык html. Разработка стиля и архитектуры Web-сайта
9. Практическая работа № 9. Язык html. Разработка стиля и архитектуры Web-сайта
10. Практическая работа № 10. Создание раздела "Библиотека". Разработка каскадных таблиц стилей (CSS)
11. Практическая работа № 11. Создание раздела "Библиотека". Разработка каскадных таблиц стилей (CSS)
12. Практическая работа № 12. Создание раздела "Библиотека". Разработка каскадных таблиц стилей (CSS)
13. Практическая работа № 13. Использование таблиц
14. Практическая работа № 14. Использование таблиц
15. Практическая работа № 15. Мультимедиа в html
16. Практическая работа № 16. Программирование в «облаке»
17. Практическая работа № 17. Программирование в «облаке»
18. Практическая работа № 18. Программирование в «облаке»

Введение

Целями освоения дисциплины являются ознакомление студентов с историей развития информационных технологий (ИТ), социальными аспектами построения информационного общества, профессиональной ответственностью и морально-этическими нормами поведения, вопросами интеллектуальной собственности и патентования, вопросами личной безопасности и свободы самовыражения в киберпространстве, влиянием ИТ на интернациональность культуры.

Методические указания к выполнению практических занятий по дисциплине «Введение в профессию» представляют следующий структурный и компонентный состав компетенций:

Знать:

- историю информационных технологий;
- влияние ИТ на социальные процессы;
- этические проблемы, возникшие с развитием ИТ;
- профессиональную ответственность и профессиональную этику ИТ;
- риски и ответственность компьютерных систем;
- содержание понятия интеллектуальной собственности, частной жизни и гражданских свобод;
- влияние ИТ на интернациональность культуры;
- содержание и состояние международного сотрудничества и в сфере ИТ;
- этические кодексы и их осуществление на практике (IEEE, ACM, SE, AITP и пр.);
- основы международного права в сфере ИТ;

Уметь:

- выявлять информационные риски и угрозы компьютерных систем;
- проводить оценку аспектов профессиональной деятельности с позиций этики;
- разрабатывать меры по управлению рисками и оценке рисков;
- учитывать их возможные последствия;

Владеть:

- методами принятия решений о законности и этичности разработки и использования ИТ в практической деятельности.

Практическая работа № 1. Применение облачных сервисов в организации сетевого взаимодействия

Аннотация

Для организации работы с облачными сервисами Google необходимо пройти процедуру регистрации аккаунта в Google.

Цель и содержание: Регистрация аккаунта в Google. Создание облачных документов на Диске Google (презентаций, таблиц, текстов, рисунков, форм).

Аппаратура и программное обеспечение.


Аппаратура. Для выполнения практической работы необходимы:

- персональный компьютер
- рабочие станции: монитор + клавиатура + мышь.


Программное обеспечение. Для выполнения лабораторной работы необходима операционная система WINDOWS 7/8/10 и подключение к глобальной сети Интернет.

Методика и порядок выполнения работы

Регистрации аккаунта в Google на Gmail.com

1. Откройте веб-браузер (Google Chrome, Internet Explorer или Mozilla Firefox)
2. В адресной строке введите адрес gmail.ru.
3. В правом верхнем углу нажмите кнопку 
4. Введите имя и фамилию **русскими** буквами.
5. Введите имя пользователя (Login). Можно использовать только **английские** буквы, цифры, точку. Этот адрес Вы будете сообщать своим корреспондентам.
6. Введите пароль. Можно использовать только **английские** буквы, цифры. Пароль должны знать только Вы.
7. Подтвердите пароль. (Снова введите этот же пароль).
8. **Запишите свой адрес (Login@gmail.com) и пароль.**
9. Введите свою дату рождения.
10. Выберите свой пол.
11. Введите номер мобильного телефона. Обратите внимание, что +7 в поле уже написано, и вам нужно написать номер, начиная с кода (922, 912, 953 и др.). (**Телефон должен быть включен и находиться под рукой**).

12. Пункт «Дополнительный адрес эл. почты» можно не заполнять

13. Докажите, что вы не робот. (Введите 2 слова, которые изображены на картинке). Если слова непонятны, нажмите кнопку  для их изменения.

14. Поставьте галочку в поле «Я принимаю Условия использования и соглашаюсь с политикой конфиденциальности Google».

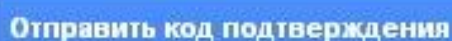


15. Нажмите кнопку

16. Подтвердите свой аккаунт:

выберите пункт «Текстовое сообщение»;

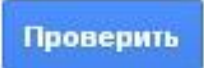
номер (если Вы его писали) должен быть заполнен;



нажмите кнопку

введите код подтверждения, который придет Вам в смссообщении на телефон;

нажмите кнопку



17. В



окне



18. нажмите кнопку



19. Нажмите кнопку

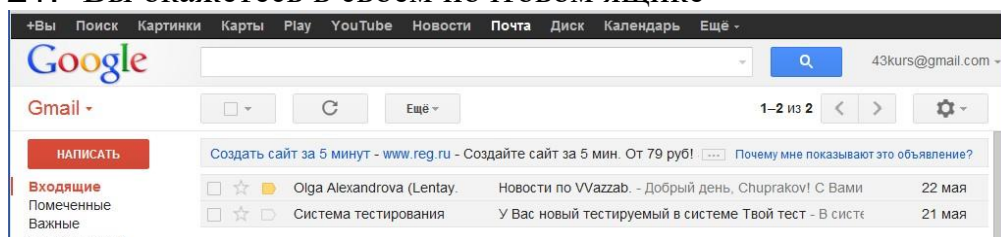
20. На странице настроек в левом верхнем углу нажмите кнопку



или



21. Вы окажетесь в своем почтовом ящике





Чтобы написать письмо, щелкните по кнопке

22. В поле «Кому»

наберите e-mail адресата **zamira.ncfu@gmail.com**

укажите тему лабораторная работа №1 задание 1, наберите текст «Я, Иванов Иван Ионович, выполнил первое задание

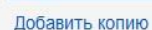






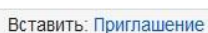




Кому

Тема



 « Обычный текст

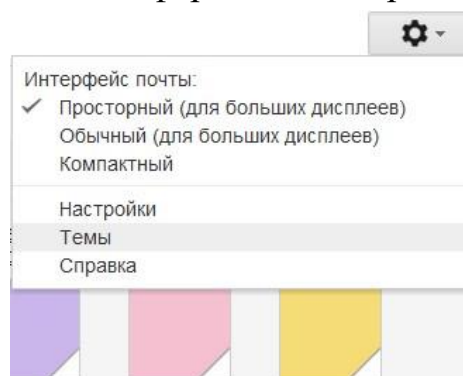
лабораторной работы № 1».

23. Чтобы отправить письмо, щелкните по кнопке

Дополнительно:

Вы можете изменить язык почтового ящика на русский.

1. нажмите кнопку  и выберите пункт **Mail settings**
2. в пункте **Gmail display language**: выберите «Русский».
3. прокрутите страницу до конца и нажмите кнопку 
4. Чтобы вставить или изменить свою фотографию, которую будут видеть Ваши адресаты, можно воспользоваться инструкцией:
<http://support.google.com/mail/bin/answer.py?hl=ru&ctx=mail&answer=35529>
5. Чтобы изменить оформление откройте пункт «Тема».



Указания по технике безопасности

При выполнении лабораторной работы **запрещается**:

1. Самостоятельно производить ремонт персонального компьютера, а так же, установку и удаление имеющегося программного обеспечения;
2. Нарушать общепринятые правила техники безопасности при работе с электрооборудованием, в частности, касаться электрических розеток металлическими предметами и т.д.;
3. Принимать пищу, напитки и сорить на рабочем месте пользователя персонального компьютера.

В случае неисправности персонального компьютера необходимо **не-**

медленно сообщить об этом обслуживающему персоналу лаборатории (системному администратору, оператору).

Содержание отчета и его форма

Отчет по лабораторной работе оформляется в виде текстового документа и должен включать:

1. Название практической работы.
2. Цель практической работы.
3. Формулировку индивидуального задания и результат его выполнения.
4. Краткие выводы по результатам выполнения практической работы.

Защита практической работы

Оформленный, в соответствии с требованиями ГОСТов, отчет представляется студентом преподавателю для проверки и последующей защиты. Защита отчета по практической работе производится студентом только индивидуально.

В ходе защите практической работы студент отвечает на вопросы преподавателя (поясняет методику выполнения индивидуального задания, отвечает на контрольные вопросы и т.д.).

Отчет, оформленный с отступлениями от требований ГОСТов, небрежно и неаккуратно к защите не принимается.

Практическая работа № 2. Использование Документов Google для организации сетевого взаимодействия

Цель и содержание: Регистрация аккаунта в Google. Создание облачных документов на Диске Google (презентаций, таблиц, текстов, рисунков, форм).

Аппаратура и программное обеспечение.

Аппаратура. Для выполнения практической работы необходимы:

- персональный компьютер
- рабочие станции: монитор + клавиатура + мышь.

Программное обеспечение. Для выполнения лабораторной работы необходима операционная система WINDOWS 7/8/10 и подключение к глобальной сети Интернет.

Методика и порядок выполнения работы

Задание:

1. Научиться использовать возможности **Документов Google** для организации сетевого взаимодействия.
2. Создать презентацию по теме «Применение Документов Google на моем занятии» и предоставить доступ к ней одногруппникам и преподавателю.
3. Создать документ-отчет по теме «Практическая работа №1», включив в него скриншоты, выполненных заданий (на которых явно виден автор работы), предоставить доступ к нему преподавателю и написать письмо преподавателю по адресу igwt@mail.ru с отчетом о выполненной первой работы.

Документы Google – это пакет продуктов, предназначенный для создания различных документов, работы над ними вместе с другими пользователями в режиме реального времени и хранения документов и других файлов в Интернете. Вы можете получить доступ к своим документам и файлам с любого компьютера в любой точке мира, если он подключен к Интернету. Получить подробные [Общие сведения о Документах Google](#) можно в «Справочном центре»

Документов

Google <http://support.google.com/drive/bin/answer.py?hl=ru&answer=49008>.

Возможности работы с документами Google.

- Преобразование файлов большинства типов в формат Документов Google.
- Оформление и форматирование документов с применением таких возможностей, как формат по образцу, поля, интервалы и шрифты.
- Приглашение других пользователей для совместной работы над документом с предоставлением им прав на редактирование, добавление комментариев или просто просмотр.
- Совместная онлайн-работа в реальном времени с возможностью общения с соавторами в чате.
 - Просмотр истории изменений документов и возврат к любой версии.
- Загрузка документов Google в форматах Word, OpenOffice, RTF, PDF, HTML и ZIP на компьютер.
 - Перевод документов на другой язык.
- Отправка документов другим пользователям по электронной почте в виде прикрепленных файлов.
- Возможность получения справки в сетевом «Справочном центре» Документов Google.

Возможности работы с таблицами.

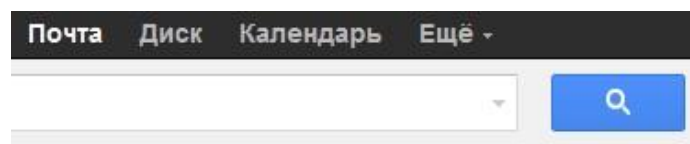
- Импорт и преобразование данных в форматах XLS, CSV, TXT и ODS.
 - Экспорт файлов в форматах XLS, CSV, TXT, ODS, PDF и HTML.
- Форматирование и изменение формул для вычисления результатов и представления данных в нужном виде.
- Общение в чате в режиме реального времени с другими пользователями, редактирующими таблицу.
 - Создание диаграмм и гаджетов.
- Встраивание таблицы или ее отдельных листов в свой блог или на веб-сайт.
- Возможность получения дополнительной информации о таблицах Google в сетевом «Справочном центре» Google.

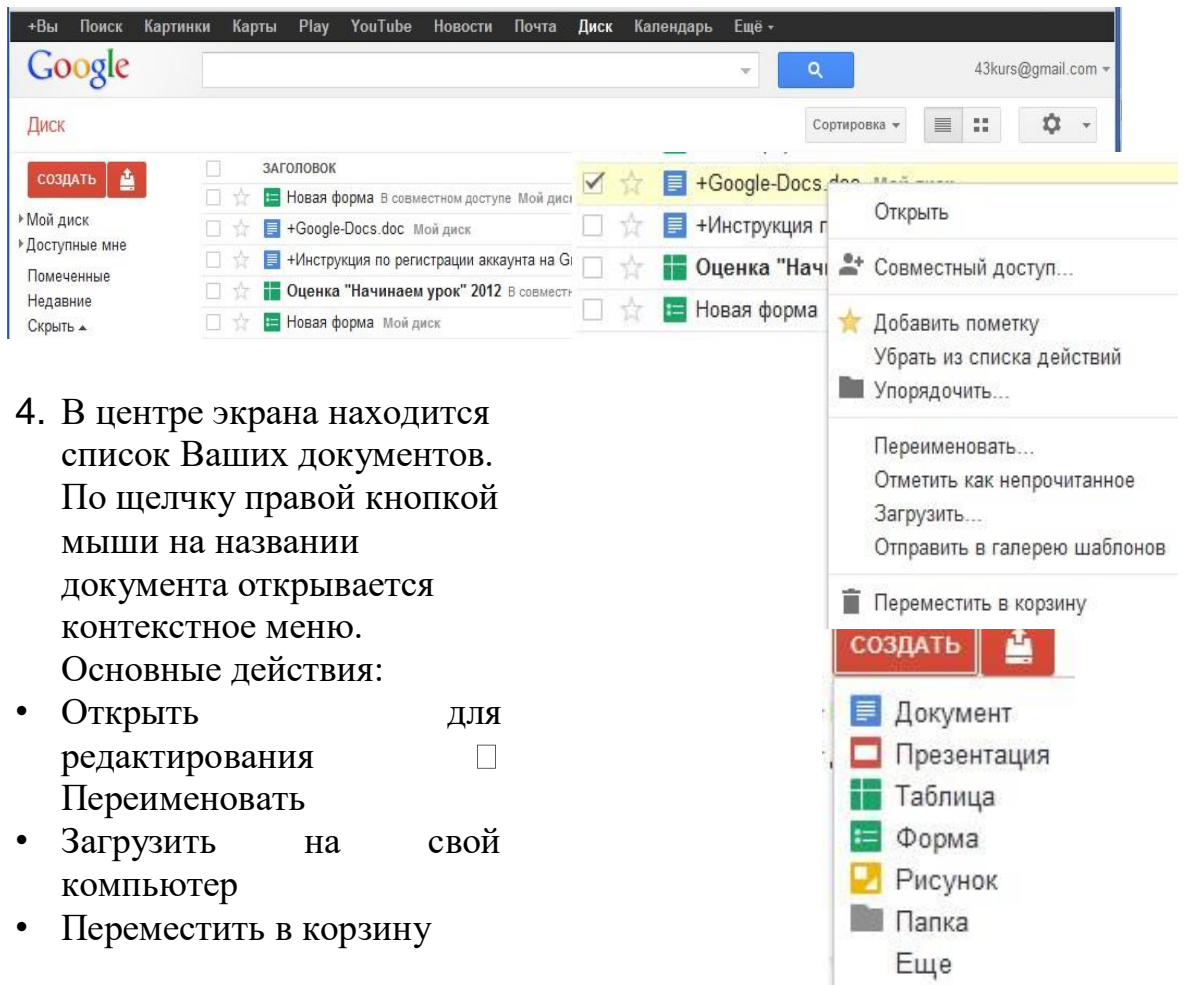
Возможности работы с презентациями.

- Общий доступ к презентациям и их редактирование совместно с друзьями и коллегами.
 - Импорт и преобразование презентаций в форматах PPT и PPS.
 - Загрузка презентаций в виде файлов в форматах PDF, PPT и TXT.
 - Вставка изображений и видео, форматирование слайдов.
 - Предоставление разрешений на просмотр презентаций в Интернете в режиме реального времени из удаленных местоположений.
 - Публикация и встраивание презентаций в веб-сайты для предоставления доступа к ним большому количеству пользователей.
 - Возможность получения справки в сетевом «Справочном центре» Документов Google.
 - Возможность получения дополнительных сведений о презентациях Google в сетевом «Справочном центре» Google.
- Возможности работы с рисунками.**
- Предоставление общего доступа к рисункам и их редактирование совместно с друзьями и коллегами.
 - Загрузка рисунков в виде файлов в форматах PNG, JPEG, SVG и PDF.
 - Вставка изображений, фигур и линий, а также их форматирование.
 - Совместная работа с другими пользователями в режиме реального времени, где бы они ни были.
 - Вставка рисунка в документ, таблицу или презентацию.
 - Возможность получения дополнительных сведений о рисунках Google в сетевом «Справочном центре» Google.

Документы Google

1. Войдите в свой аккаунт Google
2. В верхней части экрана нажмите **Диск** или **Документы**. Если этого пункта нет, откройте пункт **Ещё**
3. Вы увидите такой экран:



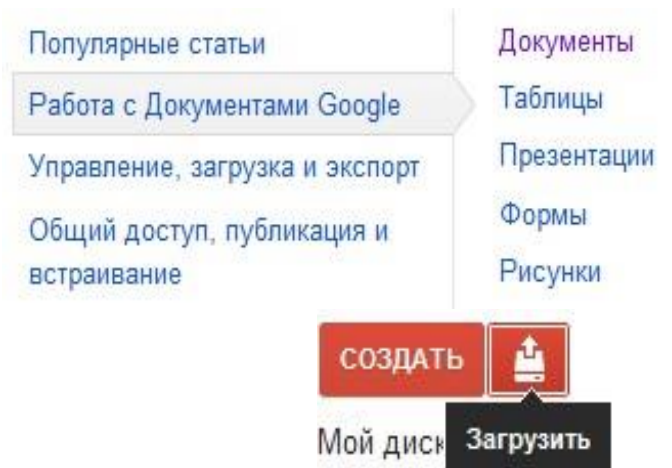


4. В центре экрана находится список Ваших документов. По щелчку правой кнопкой мыши на названии документа открывается контекстное меню. Основные действия:

- Открыть для редактирования
- Переименовать
- Загрузить на свой компьютер
- Переместить в корзину

5. Вы можете СОЗДАТЬ новый документ:

- Текстовый документ(аналог Word)
- Презентацию(аналог PowerPoint)
- Электронную таблицу(аналог Excel)
- Форму
- Рисунок



Подробная инструкция по работе с документами находится на сайте

Google(<http://support.google.com/drive/?hl=ru>)

6. Вы можете ЗАГРУЗИТЬ документ (файл) со своего компьютера в Интернет:

**Загрузить – Файл – находим
нужный файл**

7. В окне настройки загрузки ставим или убираем галочки.

- *Преобразовывать документы в формат Google*: ставим галочку, если хотим редактировать документ средствами Google (при этом некоторые эффекты будут урезаны)
Если галочку убрать, документ загрузится без изменений, его можно будет просматривать, но не редактировать.
- *Преобразовывать текст из PDF и изображений в формат документов Google* – ставим галочку, если нам нужно распознать текст (аналог FineReader)

Настройки загрузки

Установите настройки для загрузки файлов. Эти настройки будут применяться ко всем файлам, загружаемым в Документы Google

Преобразовывать документы, презентации, таблицы и рисунки в формат Документов Google

Преобразовывать текст из PDF-файлов и изображений в формат Документов Google

Язык документа:

Подтверждать настройки перед каждой загрузкой

Начать загрузку

Отмена

8. После завершения загрузки (100%) документ появится в списке

<input type="checkbox"/>	ЗАГОЛОВОК	ВЛАДЕЛЕЦ	ПОСЛЕДНЕЕ ИЗМЕНЕНИЕ
<input type="checkbox"/> ☆	programma_IKT.doc Мой диск	я	2:28 pm я
<input type="checkbox"/> ☆	Новая форма В совместном доступе Мой диск	я	13:50 я
<input type="checkbox"/> ☆	+Google-Docs.doc Мой диск	я	13:49 я
<input type="checkbox"/> ☆	+Инструкция по регистрации аккаунта на Gmail.doc Мой ди я	я	13:48 я
<input type="checkbox"/> ☆	Оценка "Начинаем урок" 2012 В с		
<input type="checkbox"/> ☆	Новая форма Мой диск		

Загрузка завершена

Настройки ▾ Преобразование: вкл.

W programma_IKT.doc Преобразование завершено [Открыть совместный доступ](#)

может скачать,
загрузить документ и
списка на свой компьютер.
Правая кнопка на
названии документа –
Загрузить
Выберите формат
документа (MSOffice,
OpenOffice, ...)
Сохраните документ в
нужном Вам месте на
своем компьютере

☆ +Google-Docs.doc Мой диск

☆ +Инструкция п

☆ Оценка "Нач

☆ Новая форма

- Открыть
- Совместный доступ...
- Добавить пометку
- Убрать из списка действий
- Упорядочить...
- Переименовать...
- Отметить как непрочитанное
- Загрузить...
- Отправить в галерею шаблонов
- Переместить в корзину

С одним документом могут работать несколько
пользователей. Для этого нужно нажать кнопку
Указываем адреса
пользователей, котор
мы предоставляем
доступ
Выбираем статус:
Редактор, Коммент
Читател
Ставим галочки

Предоставить доступ

Настройки совместного доступа

У кого есть доступ

<input type="checkbox"/>	Личный – доступ имеют только перечисленные ниже пользователи	Изменить...
<input checked="" type="checkbox"/>	Николай Чупраков (вы) 43kurs@gmail.com	Владелец

Подробная инструкция
использованию
совместного доступа к

Добавить пользователей: [Выбрать в контактах](#) Редактор ▾

Оповестить по электронной почте - [Добавить сообщение](#)

Отправить мне копию Прикрепить документ к

находится на сайте Google

Указания по технике безопасности

При выполнении лабораторной работы запрещается:

1. Самостоятельно производить ремонт персонального компьютера, а так же, установку и удаление имеющегося программного обеспечения;

2. Нарушать общепринятые правила техники безопасности при работе с электрооборудованием, в частности, касаться электрических розеток металлическими предметами и т.д.;

3. Принимать пищу, напитки и сорить на рабочем месте пользователя персонального компьютера.

В случае неисправности персонального компьютера необходимо **немедленно** сообщить об этом обслуживающему персоналу лаборатории (системному администратору, оператору).

Содержание отчета и его форма

Отчет по практической работе оформляется в виде текстового документа и должен включать:

1. Название практической работы.
2. Цель практической работы.
3. Формулировку индивидуального задания и результат его выполнения.
4. Краткие выводы по результатам выполнения практической работы.

Защита практической работы

Оформленный, в соответствии с требованиями ГОСТов, отчет представляется студентом преподавателю для проверки и последующей защиты. Защита отчета по практической работе производится студентом только индивидуально.

В ходе защите работы студент отвечает на вопросы преподавателя (поясняет методику выполнения индивидуального задания, отвечает на контрольные вопросы и т.д.).

Отчет, оформленный с отступлениями от требований ГОСТов, небрежно и неаккуратно к защите не принимается.

Практические работы № 3-4. Создание анкет с помощью Документов Google. Создание анкеты «Акклиматизация студентов первокурсников в СКФУ»

Цель и содержание: Создание анкет в Google.

Аппаратура и программное обеспечение.

Аппаратура. Для выполнения лабораторной работы необходимы:

- персональный компьютер

- рабочие станции: монитор + клавиатура + мышь.

Программное обеспечение. Для выполнения практической работы необходима операционная система WINDOWS 7/8/10 и подключение к глобальной сети Интернет.

Методика и порядок выполнения работы

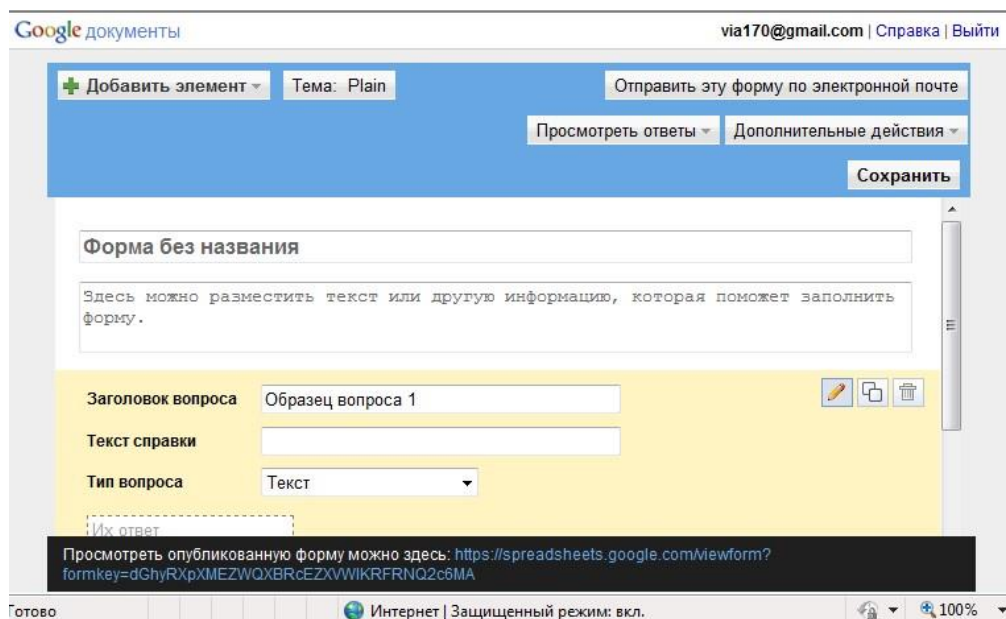
Задание:

1. С помощью сервиса Документы Гугл создать анкету для студентов, преподавателей или родителей, содержащую не менее семи вопросов, используя все типы вопросов.
2. Отправить форму по электронной почте, провести анкетирование, получить ответы (не менее 3 респондентов).
3. Предоставить доступ к таблице с результатами анкетирования преподавателю или по целевому назначению.

Сервис Документы Гугл позволяет создавать, публиковать и обрабатывать анкеты для проведения сетевых опросов. Анкету в данном сервисе называют **ФОРМОЙ**. Формы можно использовать для тестирования, но правильность ответов придется проверять преподавателю.

1. Откройте **Документы Гугл** (через свой аккаунт или Электронную почту Гугл).

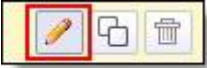

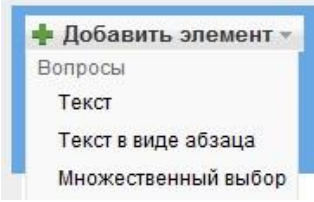
Нажмите **Создать** □ **Форма**.



2. Откроется шаблон формы.
3. В открывшемся шаблоне вместо слов *Новая форма* впишите название анкеты, ниже разместите общую информацию об анкете или небольшую инструкцию по ее заполнению.
4. Начните писать первый вопрос. Для него уже появилась заготовка.
5. В поле **Заголовок вопроса** впишите сам вопрос.
6. Поле **Пояснение** является необязательным, но оно может содержать инструкцию по ответу на Ваш вопрос. Например, к вопросу «*ФИО*» может быть инструкция «*Укажите фамилию, а также без сокращений имя и отчество. Например, Иванов Иван Иванович.*».
7. Далее выберем тип вопроса:

- **текст** – создаст небольшое текстовое поле, в которое респондент будет вписывать краткий ответ;
- **текст (абзац)** – создаст текстовое поле, в которое можно писать развернутый ответ;
- **один из списка** – создает вопрос с возможностью выбора **ОДНОГО** варианта из нескольких;
- **несколько из списка** – вопрос с возможностью выбора нескольких вариантов из предложенных;

- **выпадающий список** – создаст раскрывающийся список, из которого респондент может выбрать только один вариант ответа;
 - **шкала** – создаст шкалу оценивания, нужно только задать границы оценочной шкалы;
 - **сетка** – та же шкала, но только в виде таблицы, т.е. есть возможность оценивания по нескольким параметрам.
8. Прежде чем нажать кнопку **Готово**, подумайте, нужно ли сделать этот вопрос обязательным? Если да, то поставьте соответствующий флажок рядом с кнопкой **Готово**.
9. В правом верхнем углу каждого вопроса находятся 3 кнопки, которые позволяют: редактировать вопрос, создать его копию и удалить вопрос.

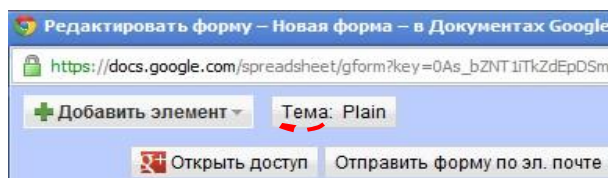
<p>Редактирование: чтобы отредактировать существующий вопрос, нажмите кнопку Изменить справа от соответствующего вопроса.</p>	
<p>Удаление: чтобы удалить вопрос, нажмите кнопку Удалить справа от соответствующего вопроса.</p>	
<p>Копирование: чтобы быстро скопировать вопрос, нажмите кнопку Скопировать справа от соответствующего вопроса.</p>	

10. Редактирование второго вопроса открывается кнопкой

Изменить. 

11. Добавить новые вопросы можно командой **Добавить элемент** в верхней части экрана.

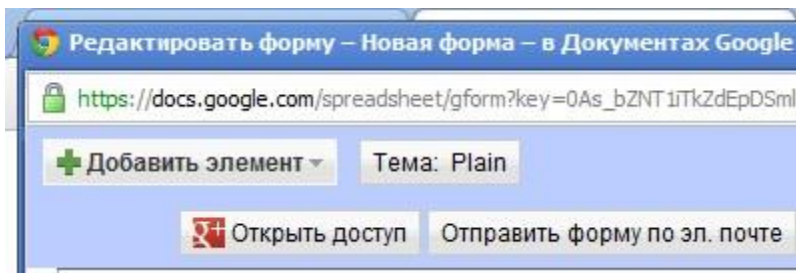
12. Форма готова, но оформление ее оставляет желать лучшего. Улучшим его, выбрав одну из готовых тем (кнопка **Тема**).



После выбора подходящего фона не забудьте нажать на кнопку **Применить** в верхней части окна. После этого изменится название темы и выбранное оформление будет

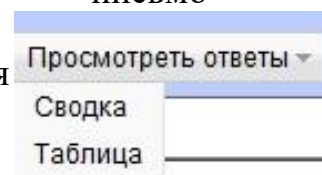
применено к готовой форме.

13. Когда анкета будет полностью готова, Вам необходимо отправить ее респондентам. Это можно сделать двумя способами:



- нажмите **Отправить форму по электронной почте** и укажите адреса электронной почты пользователей, которым требуется отправить данную форму;
- скопируйте ссылку, расположенную в черном прямоугольнике в нижней части экрана после слов *Просмотреть опубликованную форму можно здесь*, и вставьте ее в письмо

этой ссылке Ваша анкета откроется для анкетирования.



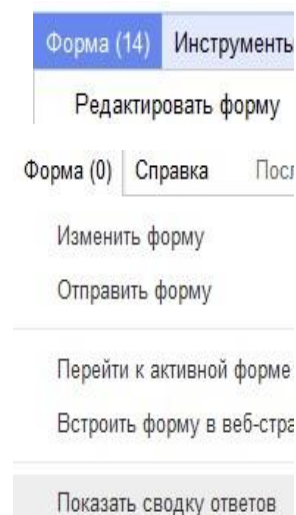
Просмотреть опубликованную форму можно здесь: <https://docs.google.com/spreadsheet/viewform?formkey=dEpDSmlBMUdmdVAzdXYakw1VEI0NUE6MQ>

или на страницу сайта или блога. По

14. Все ответы участников анкетирования будут автоматически собраны в таблицу, которая доступна из Документов Гугл. Открыть таблицу с результатами анкетирования можно двумя способами:
- на своей страничке документов Google найти анкету и открыть её.
 - в режиме редактирования формы перейти в таблицу
15. Здесь можно предоставить доступ к результатам другим

пользователям  **Предоставить доступ**

16. Если вы хотите отредактировать вопросы анкеты (перейти в режим редактирования формы) нажмите кнопку **«Форма»** → **«Редактировать форму»**. Далее вы можете изменять вопросы, редактировать их или копировать.



17. Можно перейти в режим **Сводка**, нажав кнопку **Показать сводку отчетов**. Здесь ответы будут

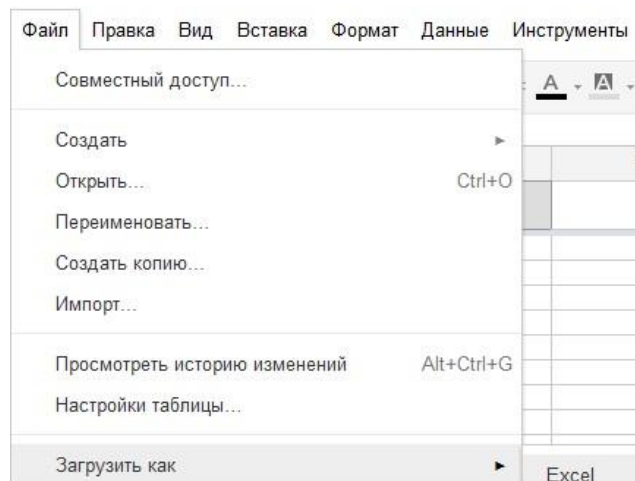
представлены в числовой форме, процентном соотношении и в виде диаграмм.



18. Также предусмотрена возможность:

- Использовать форматирование и редактор формул в таблицах, чтобы подсчитывать результаты и настраивать вид данных.
- Импортировать и экспортировать данные в формате XLS, CSV, TXT и ODS (и экспортировать в формате PDF и HTML).

Например, можно загрузить таблицу на свой компьютер и работать с ней в программе Excel.



Темы для опросов:

1. Опрос по программному обеспечению ПК
2. Опрос по аппаратному обеспечению ПК
3. Опрос по операционным системам
4. Опрос по технологиям защиты информации

5. Опрос по качеству обучения по различным дисциплинам на курсе
6. Опрос по сетевым технологиям
7. Опрос по робототехнике
8. Опрос по социальным сетям
9. Опрос по базам данных
10. Опрос по web-технологиям

Задание 2. Составить анонимный опрос по теме «Акклиматизация студентов первокурсников в СКФУ»

Указания по технике безопасности

При выполнении практической работы **запрещается**:

1. Самостоятельно производить ремонт персонального компьютера, а так же, установку и удаление имеющегося программного обеспечения;
2. Нарушать общепринятые правила техники безопасности при работе с электрооборудованием, в частности, касаться электрических розеток металлическими предметами и т.д.;
3. Принимать пищу, напитки и сорить на рабочем месте пользователя персонального компьютера.

В случае неисправности персонального компьютера необходимо **немедленно** сообщить об этом обслуживающему персоналу лаборатории (системному администратору, оператору).

Содержание отчета и его форма

Отчет по практической работе оформляется в виде текстового документа и должен включать:

1. Название практической работы.
2. Цель практической работы.
3. Формулировку индивидуального задания и результат его выполнения.
4. Краткие выводы по результатам выполнения практической работы.

Защита практической работы

Оформленный, в соответствии с требованиями ГОСТов, отчет представляется студентом преподавателю для проверки и последующей защиты. Защита отчета по практической работе производится студентом только индивидуально.

В ходе защите работы студент отвечает на вопросы преподавателя (поясняет методику выполнения индивидуального задания, отвечает на контрольные вопросы и т.д.).

Отчет, оформленный с отступлениями от требований ГОСТов,

небрежно и неаккуратно к защите не принимается.

Практическая работа № 5. Оценка качества Интернет ресурса

Цель и содержание: научиться оценивать по необходимым характеристикам качество Интернет ресурса.

Аппаратура и программное обеспечение.

Аппаратура. Для выполнения лабораторной работы необходимы:

- персональный компьютер

- рабочие станции: монитор + клавиатура + мышь.

Программное обеспечение. Для выполнения практической работы необходима операционная система WINDOWS 7/8/10 и подключение к глобальной сети Интернет.

Методика и порядок выполнения работы

1. Изучить лекционный материал

Существуют четыре основные причины, по которым пользователи на одни сайты возвращаются, а на другие - нет. Эти четыре фактора - основа хорошего веб-дизайна, поскольку именно этого больше всего хотят пользователи. Они могут быть собраны воедино с помощью аббревиатуры HOME:

- Высокое качество содержания (High-quality content)
- Частые обновления (Often update)
- Минимальное время загрузки (Minimal download time)
- Простота использования (Ease of use)

Чтобы перейти от HOME-дизайна к дизайну "HOME-RUN", добавьте три дополнительных качества:

- Соответствие потребностям пользователя (Relevant to user`s needs)
- Уникальность в Интернете (Unique to the online medium)
- Ориентированная на Интернет корпоративная культура (Net-centric corporate culture)

Концепция высокого качества веб-страниц общего характера сформулировать:

- красивая форма дизайна, графика;
- умное использование HTML кода и подключаемых модулей, которые делают сайт интерактивным и удобным в пользовании;
- наличие инновационных материалов;
- наглядность, интуитивность интерфейса

Судить о качестве Интернет ресурса и его предназначенности для своих профессиональных целей следует по нижеследующим признакам.

URL домены

Каждый унифицированный указатель информационного ресурса (URL) заканчивается суффиксом, который указывает на происхождение и назначение сайта:

- *.com Коммерция
- *.org Международная организация

- *.gov Государственный департамент
- *.edu Образование
- *.mil Военное дело
- и т.д.

Например:

- whitehouse.gov- сайт правительства США, президентский сайт
- whitehouse.org - пародия на государственный сайт
- whitehouse.com - порно сайт

Авторы

Сведения об авторе, ответственном лице, адрес e-mail и другие сведения об "ответственности" обычно располагаются внизу страницы. Следует иметь в виду, что даже страницы солидных профессионалов могут иметь субъективные мнения.

Библиография

Научные статьи или веб-страницы научных или образовательных ресурсов могут включать библиографию печатных работ. Убедитесь, что это не самоцитирование (автора или источника), включены библиографические сведения о научных книгах, а не только популярные источники, актуальны ли использованные источники? Высокое качество библиографических записей указывает на высокий качественный сайт.

Дата

Сайт может хорошо выглядеть, но если дата его последнего обновления устарела на год, то и информация может быть устаревшей. В зависимости от темы исследования, результаты последних исследований и нынешние события могут оказаться решающими. Это особенно верно, когда речь идет о научно-технических и экономических вопросах.

Гиперссылки

Гиперссылки должны дополнять страницу, добавив контекстную поддержку, более детальную информацию, альтернативную точки зрения, смежные темы.

Проверьте надежность гиперссылок:

Если они укажут вам на другие научные ресурсы и профессиональных организаций, то страницы, которые вы используете, авторитетны, если они ведут к коммерческим сайтам, то надо быть осторожными. Большое количество "мертвых" ссылок указывают на не надежный ресурс.

Внутренние доказательства

Определена ли целевая аудитория и задачи ресурса. Они соответствуют вашим ожиданиям? Изучите название и краткое содержание (если таковое предусматривается), основные идеи, содержание, стиль изложения и научность, профессиональная точность, грамотность.

Доступность

Хороший ресурс доступен во всех своих частях. Барьеры к доступу могут создавать следующие элементы:

- специализированное программное обеспечение и модули подключения,
- использование графических или звуковых файлов,
- формат нецензурных и жаргонных выражений
- вознаграждения или специальной регистрации.

Будьте внимательны по отношению к любому объекту, который требует паспортных данных или любой другой конфиденциальной информации о пользователе кредитной карты.

Оформление

Общее оформление сайта должен отражать качественный веб дизайн: графический дизайн, навигация, доступность. Возможность избежать электронной почты и дискуссионных групп.

2. Выполнить задание.

2.1. Протестировать предложенные веб-страницы:

<http://ncfu.ru/>

<http://www.gmf.ru:9002/manage/page>

<http://www.management.com.ua/bp/bp023.html>

<http://www.sci.aha.ru/>

<http://www.nstu.ru>

<http://gov.ru>

<http://www.ru.emb-japan.go.jp/>

<http://www.spsl.nsc.ru/>

<http://www.tomsk.ru/>

<http://www.ras.ru/>

<http://www.hse.ru/>

<http://www.stanford.edu/>

<http://www.cfin.ru/>

<http://www.berkeley.edu/>

<http://www.philosophy.ru/>

Порядок выполнения:

1. Запустить браузер Интернет (любой)
2. Зайти на страницы веб-ресурсов, предложенные в задании
3. Оценить качество каждого ресурса по показателям:

Достоверность Web ресурса

Точность

Управление

Авторитетность

Объективность

Оперативность

Актуальность

Удобство

Доступность

Сочетание всех параметров

4. Заполнить таблицу "Качество веб-ресурса": проставить рейтинг сайтов.
5. Создать отчет о выполнении лабораторной работы

Указания по технике безопасности

При выполнении практической работы **запрещается**:

1. Самостоятельно производить ремонт персонального компьютера, а так же, установку и удаление имеющегося программного обеспечения;
2. Нарушать общепринятые правила техники безопасности при работе с электрооборудованием, в частности, касаться электрических розеток метал-

лическими предметами и т.д.;

3. Принимать пищу, напитки и сорить на рабочем месте пользователя персонального компьютера.

В случае неисправности персонального компьютера необходимо **немедленно** сообщить об этом обслуживающему персоналу лаборатории (системному администратору, оператору).

Содержание отчета и его форма

Отчет по практической работе оформляется в виде текстового документа и должен включать:

1. Название практической работы.
2. Цель практической работы.
3. Формулировку индивидуального задания и результат его выполнения.
4. Краткие выводы по результатам выполнения практической работы.

Защита практической работы

Оформленный, в соответствии с требованиями ГОСТов, отчет представляется студентом преподавателю для проверки и последующей защиты. Защита отчета по практической работе производится студентом только индивидуально.

В ходе защите практической работы студент отвечает на вопросы преподавателя (поясняет методику выполнения индивидуального задания, отвечает на контрольные вопросы и т.д.).

Отчет, оформленный с отступлениями от требований ГОСТов, небрежно и неаккуратно к защите не принимается.

Практические работы № 6-8. Программирование в «облаке»

Цель работы:

Познакомиться с основными особенностями языка C#. Изучить схему работы .Net и C#. Рассмотреть листинги элементарных программ на языке C#. Подготовить отчет по выполненным заданиям.

Аппаратура и программное обеспечение.

Аппаратура. Для выполнения практической работы необходимы:

- персональный компьютер
- рабочие станции: монитор + клавиатура + мышь.

Программное обеспечение. Для выполнения лабораторной работы необходима операционная система WINDOWS 7/8/10 и подключение к глобальной сети Интернет.

Методика и порядок выполнения работы

Теоретический материал

В этой работе показано, как можно использовать облачные вычисления при обучении основам программирования. Приводится подробный пример работы с Web-сервисом, позволяющим создавать и отлаживать учебные программы на любом языке программирования с помощью облачного сервиса сайта <http://ideone.com>.

Современная практика программирования предполагает активное использование специализированных интегрированных средств разработки (IDE – Integrated Development Environment). Их использование связано со следующими двумя сложностями:

- Настройка и установка IDE требует высокой квалификации системного администратора.
- Современные IDE достаточно требовательны к ресурсам вычислительной машины, на которой они используются.

Поясним каждый пункт подробно. Для обеспечения полнофункциональной работы IDE требуется, чтобы квалификация системного администратора, осуществляющего установку, настройку и поддержку IDE была достаточно высока. Это приводит к необходимости нанимать в учебные заведения на должность системного администратора высококвалифицированных сотрудников, заработная плата которых может оказаться существенной статьёй расходов в бюджете образовательного учреждения.

Более того, затраты образовательного учреждения могут возрасти вследствие того, что современные IDE требуют наличия высокопроизводительных вычислительных машин. Например, одна из самых распространенных IDE Microsoft Visual Studio 2012 требует для нормальной работы процессор мощностью 1,6 ГГц или выше, 1 ГБ ОЗУ (или

1,5 ГБ для виртуальной машины), 10 ГБ свободного дискового пространства [7]. Для большинства задач образовательных учреждений не требуется компьютеров с такой высокой производительностью, поэтому их покупка может оказаться недопустимой роскошью.

Обе указанные проблемы позволяет решить применение облачных технологий при обучении программированию. В настоящее время существуют большое количество так называемых онлайн-IDE, которые не требуют установки на компьютер пользователя и которые требуют для запуска лишь наличие Интернет-браузера. Системные требования браузеров к оборудованию вычислительной машины традиционно являются скромными. Например, популярный Веб-браузер Mozilla Firefox 17 требует для установки процессор от 1300 МГц, 512 МБ ОЗУ и 200 МБ свободного дискового пространства [6], что существенно меньше приведённых ранее цифр для IDE Microsoft Visual Studio 2012.

Рассмотрим ниже, как можно использовать онлайн-IDE в учебных заведениях для обучения основам программирования на примере <http://ideone.com>. Этот сервис позволяет в режиме онлайн создавать тексты программ на разных языках программирования и запускать эти программы на исполнение с возможностью анализа полученных результатов. Основные рабочие элементы Ideone показаны на [рис. 2.1](#).

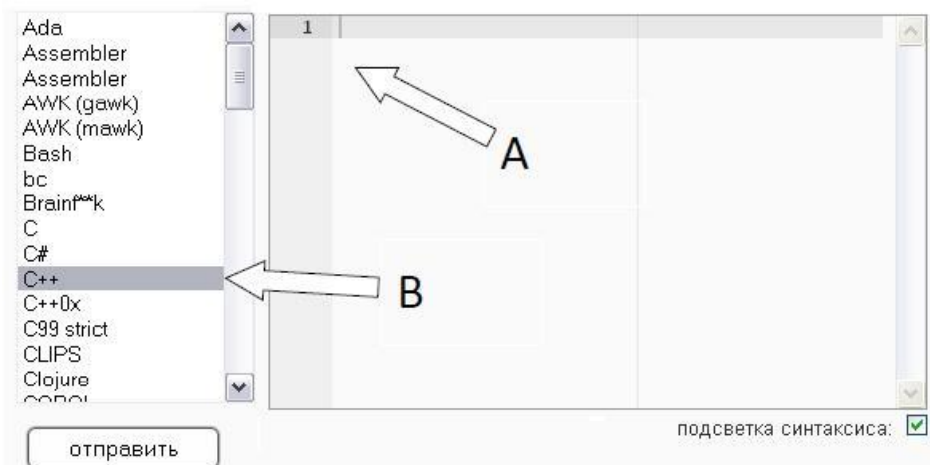
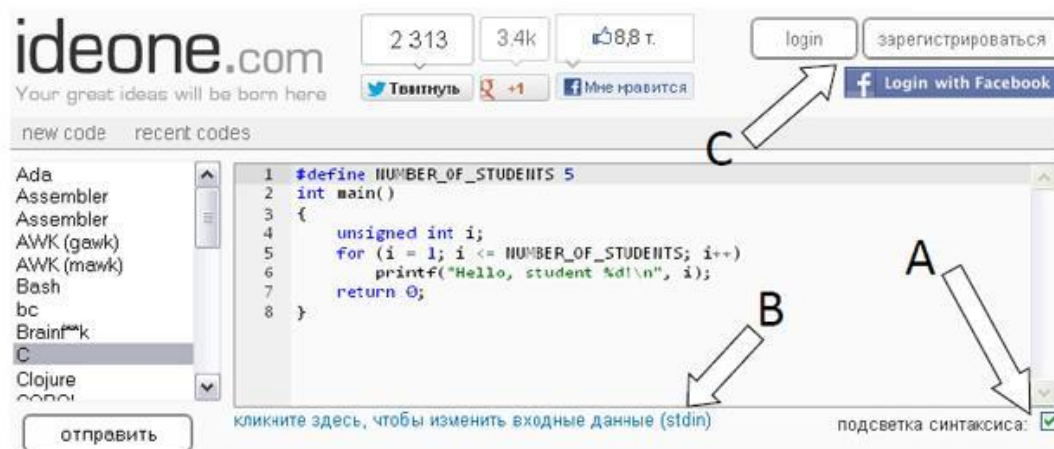


Рис. 2.1. Основные рабочие элементы Ideone

В поле "А" необходимо ввести текст программы, а в поле "В" нужно выбрать используемый язык программирования, затем нужно нажать кнопку "Отправить". В [2] указано, что Ideone поддерживает работу со следующими 55 популярными языками программирования: Ada, Assembler, AWK, Bash, bc, Brainf**k, C, C#, C++, C++0x, C99 strict, CLIPS, Clojure, COBOL, Common Lisp(clisp), D (dmd), Erlang, F#, Factor, Falcon, Forth, Fortran, Go, Groovy, Haskell, Icon, Intercal, Java, JavaScript, Lua, Nemerle, Nice, Nimrod, Node.js, Objective-C, Ocaml, Oz, PARI/GP, Pascal, Perl, PHP, Pike, Prolog, Python, R, Ruby, Scala, Scheme (guile), Smalltalk, SQL, Tcl, Text, Unlambda, VB.NET, Whitespace. Очевидно, что этого перечня достаточно при обучении основам программирования практически в любом учебном заведении мира. Более того, при использовании сервиса ideone.com у преподавателя появляется возможность использовать при обучении сразу несколько языков программирования без необходимости поддерживать работу нескольких IDE.

Покажем на примере, как может быть организована работа в группе при обучении основам программирования. [Рис 2.2](#) иллюстрирует способ запуска простой программы на языке Си. Как можно видеть, в тексте программы используется подсветка синтаксиса, аналогичная той, что пользователи привыкли использовать в обычных офлайн-IDE.

Однако при желании подсветка может быть отключена с помощью элемента управления "А". С помощью элемента управления "В" можно указать перечень входных данных для программы, что позволяет реализовать более сложную логику работы программы, чем в приведённом примере.



увеличить изображение
 Рис. 2.2. Запуск программы в Ideone

Очень важным является элемент управления "С", который позволяет персонализировать работу с программой. Данная возможность крайне ценна при организации учебного процесса. Если преподаватель попросит всех студентов зарегистрироваться в Ideone (или использовать для входа свою учётную запись Facebook), то появляется возможность сделать процесс работы с программой коллективным, а процесс совместной работы с программой будет проходить с использованием современных технологий Web 2.0. Подробнее об этом расскажем, используя [рис. 2.3](#).

[Рис 2.3](#) представляет собой результаты запуска программы, приведённой на [рис. 2.2](#), самые важные из которых помечены знаком "А". Это консольный вывод программы и возвращаемое значение. Знаком "В" отмечена Интернет-ссылка, которую преподаватель может переслать студентам для ознакомления с результатам работы демонстрационной программы, либо сами студенты могут выслать подобную ссылку преподавателю в качестве отчёта о проделанной работе.

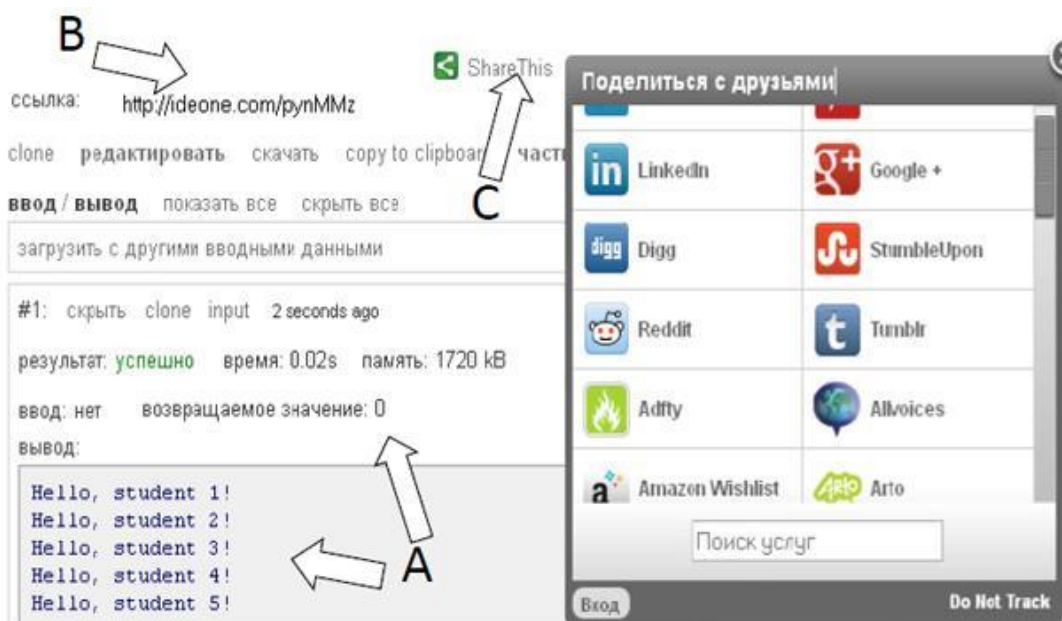


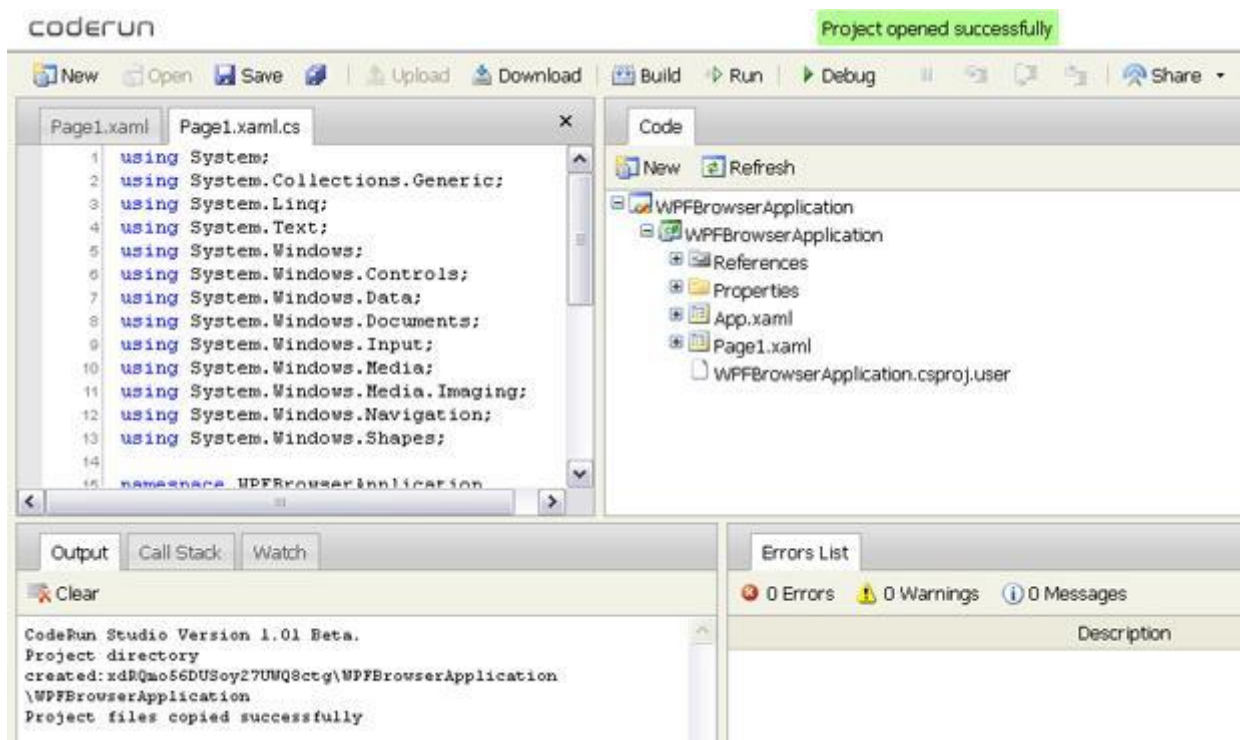
Рис. 2.3. Результаты работы программы в Ideone

Следующим этапом совместной работы может стать онлайн-обсуждение результатов работы программы с помощью средств Web 2.0 одного из популярных сервисов социальных сетей. Это становится возможным благодаря использованию элемента управления "С" на [рис. 2.3](#).

Этот элемент управления позволяет выбрать из списка в правой части экрана один виджетов популярных сайтов социальных сетей.

К сожалению, онлайн-IDE Ideone позволяет реализовать не все из функций традиционных офлайн-IDE. Например, отсутствует возможность использовать функции работы с сетью, обращения к файлам и некоторые другие. Также невозможно запустить программу, время выполнения которой займёт более 15 секунд или потребности в оперативной памяти превысят 256 МБ, или объём программы превысит 64 КБ [2]. Все эти ограничения являются достаточно серьёзными, если планируется использовать Ideone для разработки профессионального программного обеспечения. Однако для образовательных целей эти ограничения более чем приемлемы. Кроме того, для более требовательных преподавателей существуют платные и бесплатные сервисы, аналогичные Ideone, которые при этом в большей степени реализуют функционал традиционных офлайн-IDE. Таким примером являются сервисы Cloud9 IDE (www.c9.io), CodeRun [3]. На [рис. 2.4](#) показан вид интерфейса сервиса CodeRun: пользователи, работавшие с традиционными офлайн-IDE, сразу увидят много знакомых элементов управления. Присутствует панель со списком используемых классов, со списком задействованных в проекте файлов, а также окна с отладочной информацией о состоянии стека вызовов подпрограмм и с ошибками времени компиляции или времени выполнения.

Созданный проект можно сохранить в офлайн, но предпочтительным является режим работы только в облаке. Все операции, включая отладочную сборку, анализ результатов выполнения в консольном режиме, компиляцию под разные платформы и операционные системы можно выполнить в режиме онлайн. В идеале окончанием работы программиста будет скачивание готовых бинарных файлов с работающей программой. Такой подход позволяет сэкономить используемой офлайн дисковое пространство, а также позволяет компилировать проект существенно более быстро, чем на рабочем месте пользователя, если это рабочее место оборудовано устаревшим аппаратным обеспечением. Итогом этого является возможная финансовая экономия для образовательного учреждения.



увеличить изображение
Рис. 2.4. Интерфейс системы Coderun

Однако помимо экономического эффекта, можно получить и существенные преимущества при организации образовательного процесса. Студенты получают возможность совместно редактировать программные проекты, находясь у себя дома. Это позволяет реализовывать сложные курсовые проекты и лабораторные работы с существенной экономией на осуществление организационных мероприятий со стороны преподавателя.

Современные языки программирования вообще и C# в частности скрывают многие низкоуровневые особенности работы компьютеров, так что детально разбираться в особенностях работы [регистров процессора](#). Но некоторые вещи стоит держать в уме.

Что такое язык программирования? Это язык, который понимает компьютер. Но на самом деле компьютер понимает только нули и единицы. Так что любой современный язык компиляции на самом деле промежуточен - после того как вы напишете программу, он переводится на настоящий язык компьютера - компилируется. Таким образом сначала мы пишем программу на языке программирования в обычном текстовом файле, потом компилируем в двоичный язык компьютера.

C# в этом плане еще сложнее - сначала ваша программа на C# переводится в специальный промежуточный язык - [байт-код](#), после чего уже этот промежуточный язык переводится в двоичный код. Промежуточный язык рассчитан на специальный набор промежуточных программ, которые надо установить на компьютер перед запуском программы на C# - [.Net Framework](#).



Схема работы .Net и C#

Таким образом мы:

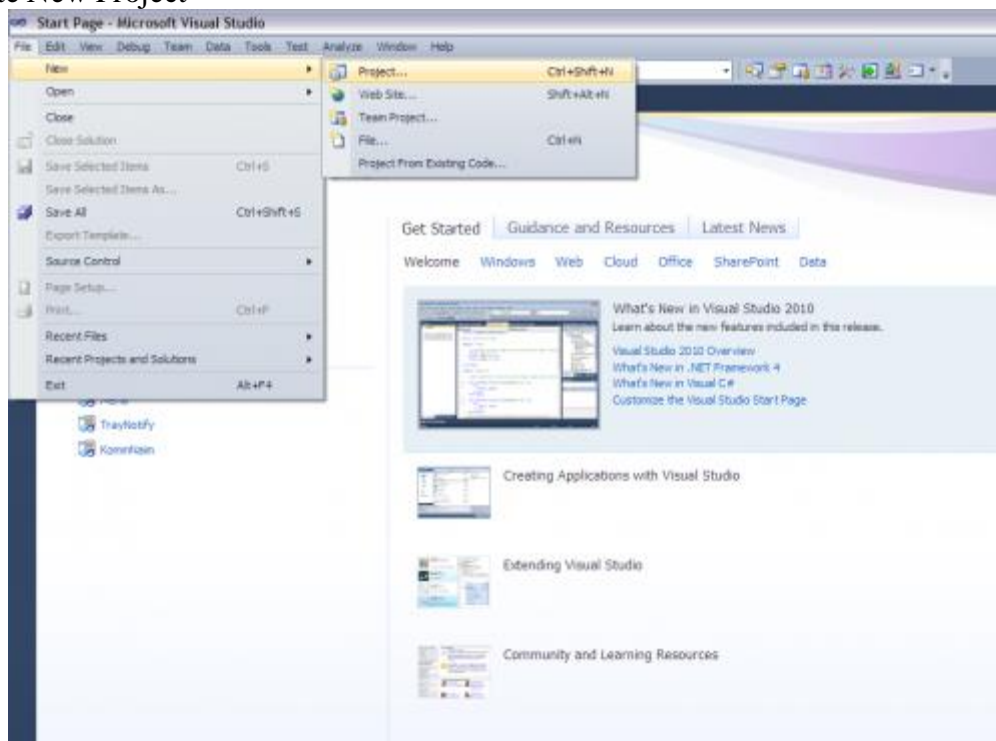
1. Пишем код на языке C# в обычный текстовый файл
2. Компилируем программу в байт-код с помощью компилятора C# (1 и 2 можно делать без Visual Studio, например набрать исходный текст в Блокноте и скомпилировать через командную строку)
3. Запускаем полученную программу на компьютере, где уже установлен .Net Framework соответствующей версии), который переводит программу в двоичный код и собственно запускает, давая доступ к компьютеру. (соответственно на компьютере без Net Framework программа не запустится вообще).

Выполнить задание:

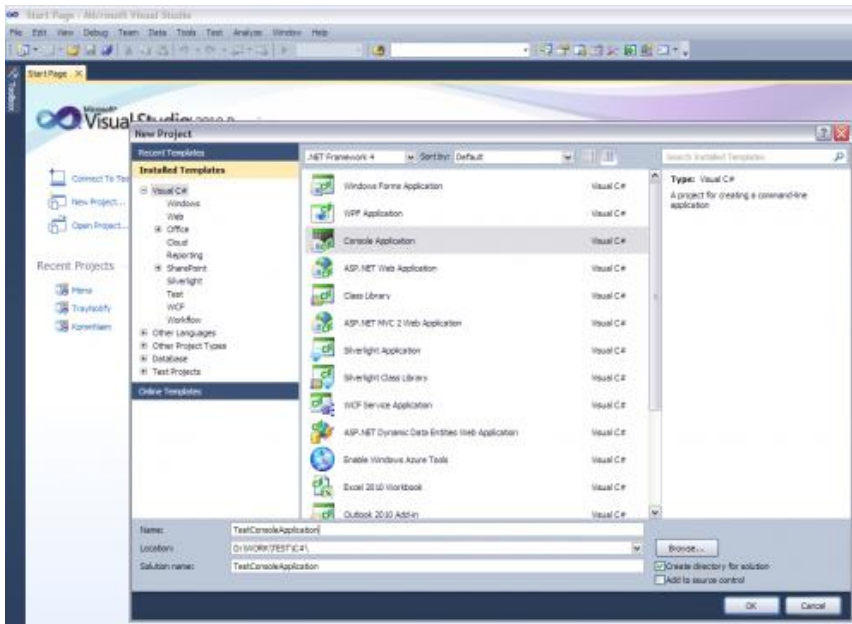
Задание 1. Привет мир!

Перейдем к практике, для создания самой простой программы мы запустим Visual Studio и выполним следующее:

1. Пойдем в меню File-New Project (Файл - Новый проект) или сразу кликнем на Create New Project



2. Выберем шаблон Console Application (Консольное приложение)



3. Выберем каталог на диске, где будет расположен наш проект

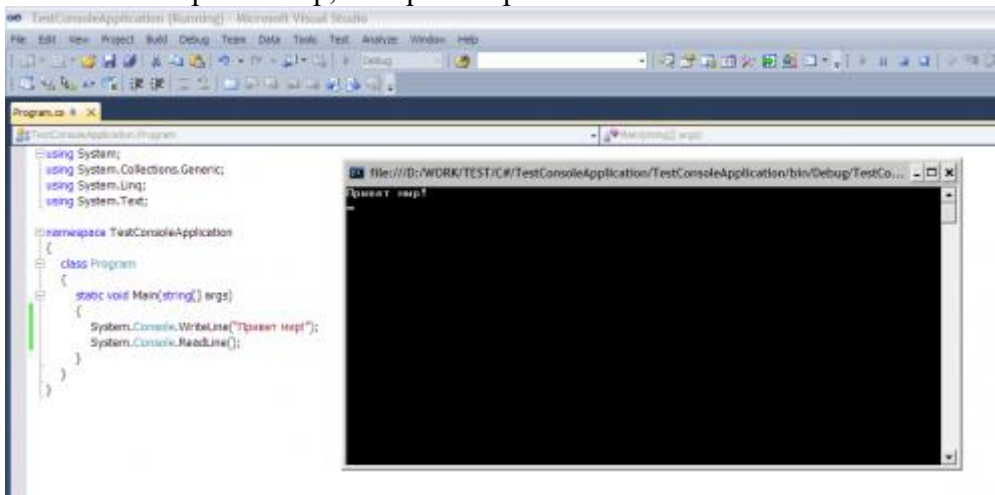
4. Добавим две строчки в текст программы, чтобы он выглядел вот так

```

static void Main(string[] args)
{
    System.Console.WriteLine("Привет мир!");
    System.Console.ReadLine();
}

```

5. Нажмем F5 (можно выбрать в меню Debug - Start Debugging) и увидим черное окошко с нашим Привет мир, которое закроется после нажатия Enter.



Что это и зачем оно надо? В теории это программа без кнопок и меню, которой можно управлять через командную строку. В наше время такие почти не используются, но знать об их существовании полезно - программисту может доводиться с такой работать или даже писать. Нам она нужна по одной-причине - такая программа меньше всего будет отвлекать от изучения универсальных особенностей языка C#.

В самом простом случае (это про нас) компьютер последовательно выполняет команды программиста - одну за другой:

System.Console.WriteLine("Привет мир!"); - выводит в консоль (то самое черное окно) строку "Привет мир"

System.Console.ReadLine(); - читает строку, которую мы вводим в командную строку и переходит к следующей команде.

Так как следующей команды нет, программа завершает работу и закрывает окно.

Задание 2. Простые переменные

Обычно программы используются для хранения и обработки какой-то информации. Например списка покупателей магазина, сотрудников фирмы или героев и монстров в компьютерной игре. При работе программы все ее данные хранятся в переменных. Переменные и сама программа хранятся в [оперативной памяти](#).

Переменную можно рассматривать как ящик или сумку для данных. Однажды сделав такой ящик мы можем класть в него разные вещи. Само собой на нижнем уровне абсолютно все данные хранятся в виде нулей и единиц, но языки высокого уровня скрывают от нас ненужные сложности. Под каждый тип данных - строку, число, картинку - нужен ящик соответствующего типа, вы ведь не будете складывать деньги, мыло и воду в одну и ту же коробку. Так что у каждой переменной есть тип, который надо указать при ее создании. Тип одновременно описывает и размер переменной - сколько данных в нее можно сложить.

Некоторые из [простых типов данных в C#](#)

- [int](#), целое число от -2,147,483,648 до 2,147,483,647 строка
- [string](#), строка
- [double](#), число с дробной частью и переменным количеством знаков после запятой, например 1.38 принимает значения от -1,79769313486232 в 308 степени до 1,79769313486232 в 308 степени. Проще говоря очень большие.
- [decimal](#), число с дробной частью и повышенной точностью расчетов, специально для финансовых операций - в миллионных и более мелких долях float могут накапливаться мелкие ошибки, банки это не любят
- [bool](#) - специальный тип для передачи данных об истинности или ложности, принимает только два значения - true или false. В старых языках вместо него использовали int с значениями 1 и 0, но это было менее наглядно и приводило к ошибкам.

И [многие другие типы](#).

Как уже говорилось выше очень часто программы обрабатывают данные о людях. Попробуем это сделать и мы.

```
Без имени - Google Chrome
about:blank

static void Main(string[] args)
{
    string stringToShow1, stringToShow2;

    string surname = "Шульженко";
    string name = "Олег";
    string otchestvo = "Васильевич";

    int age = 40;
    double weight = 88.73;

    stringToShow1 = surname + " " + name + " " + otchestvo + ", возраст " + age + ", вес " + weight;

    surname = "Чугунов";
    name = "Александр";
    otchestvo = "Игоревич";

    age = 23;
    weight = 66;

    stringToShow2 = surname + " " + name + " " + otchestvo + ", возраст " + age + ", вес " + weight;

    System.Console.WriteLine(stringToShow1);
    System.Console.WriteLine(stringToShow2);

    System.Console.ReadLine();
}
```

Не забудьте запустить по F5 и посмотреть результаты.

Здесь мы объявляем две переменные-строки, оставляем их пока пустыми, объявляем набор переменных с данными о людях, склеиваем из них одну большую строку и заносим в первую переменную-строку, заносим в тот же самый набор переменных другие данные, склеиваем все вместе и заносим в другую строку и только после этого выводим две финальные строки на экран.

Задание 3. Арифметические операции

Компьютеры почти всегда что-то считают. Собственно само слово [computer означает вычислитель](#). В этой области даже самый слабенький процессор намного превосходит человеческий мозг.

```
Без имени - Google Chrome
about:blank

static void Main(string[] args)
{
    int a = 5;
    int b = 2;
    System.Console.WriteLine("a = " + a + ", b = " + b);

    int result = a + b;
    System.Console.WriteLine("Сложение, a + b = " + result);

    result = a * b;
    System.Console.WriteLine("Умножение, a * b = " + result);

    result = a / b;
    System.Console.WriteLine("Деление, a / b = " + result + " а и b - целые числа, деление только нацело");

    double resultDouble = a / b;
    System.Console.WriteLine("Деление, a / b = " + resultDouble + " все равно что-то не так...");

    double aDouble = 5;
    resultDouble = aDouble / b;
    System.Console.WriteLine("Деление, a / b = " + resultDouble);

    System.Console.ReadLine();
}
```

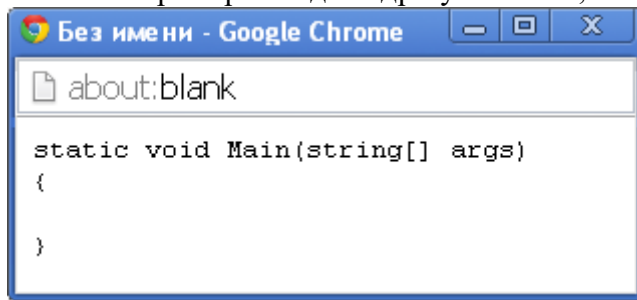
Почему произошло столь странное недоразумение с дробным числом во второй раз? Все дело в том, что внутри одной строки мы выполнили сразу несколько операций. А в каком порядке они выполняются? В порядке приоритета. И у оператора присваивания =, который записывает данные в переменную, этот приоритет один из самых низких

Сначала выполнилось сложение двух целочисленных переменных a и b и только после этого целый результат деления записался в дробную переменную doubleResult.

Небольшое

отступление:

В дальнейших примерах кода подразумевается, что вы самостоятельно вставите их внутрь

A screenshot of a Google Chrome browser window. The title bar reads "Без имени - Google Chrome". The address bar shows "about:blank". The main content area displays a C# code snippet:

```
static void Main(string[] args)
{
}

```

Задание 4. Управление и логика

На данный момент мы просто выполняем команды по очереди, одну за другой, а хотелось бы принимать решения в зависимости от внешних условий - например "если есть возможность, выполнить действие, иначе работать над созданием такой возможности". Делается это с помощью специальной конструкции если-иначе, или, на родном для C# английском [if-else](#) и специальных операторов сравнения:

[<](#) - величина слева меньше

[>](#) - величина слева больше

[<=](#) и [>=](#) - меньше либо равно и больше либо равно

[==](#) - просто равно, обратите внимание, что равенство очень похоже на оператор присваивания, который записывает значение в переменную = - не путайте их, равенство - это двойное "равно"

[!=](#) - не равно

[и другие](#)

Компьютерная логика проста - высказывание или результат сравнения может быть либо истиной либо ложью, либо true либо false, либо уничтожить человечество либо нет. Третьего компьютеру не дано. Для хранения таких данных существует специальный тип данных - логический - [bool](#) - который может принимать только два значения.

Нам важно понять, что любая операция сравнения, например $2 < 4$ на самом деле вычисляет логическое значение типа bool, которое может принимать только два значения. И уже им будет руководствоваться компьютер, выполняя нашу программ.

```
about:blank

bool boolVariable = true;

if (boolVariable)
{
    System.Console.WriteLine("boolVariable = true; --- Истина!");
}
else
{
    System.Console.WriteLine("boolVariable = false; Ложь!");
}

System.Console.WriteLine();

boolVariable = false;

if (boolVariable)
{
    System.Console.WriteLine("boolVariable = false; Истина!");
}
else
{
    System.Console.WriteLine("boolVariable = false; Ложь!");
}

boolVariable = 2 < 4;

if (boolVariable)
{
    System.Console.WriteLine("boolVariable = 2 < 4; Истина!");
}
else
{
    System.Console.WriteLine("boolVariable = 2 < 4; Ложь!");
}

if (10 != 100)
{
    System.Console.WriteLine("10 != 100! Ваш капитан очевидность!");
}

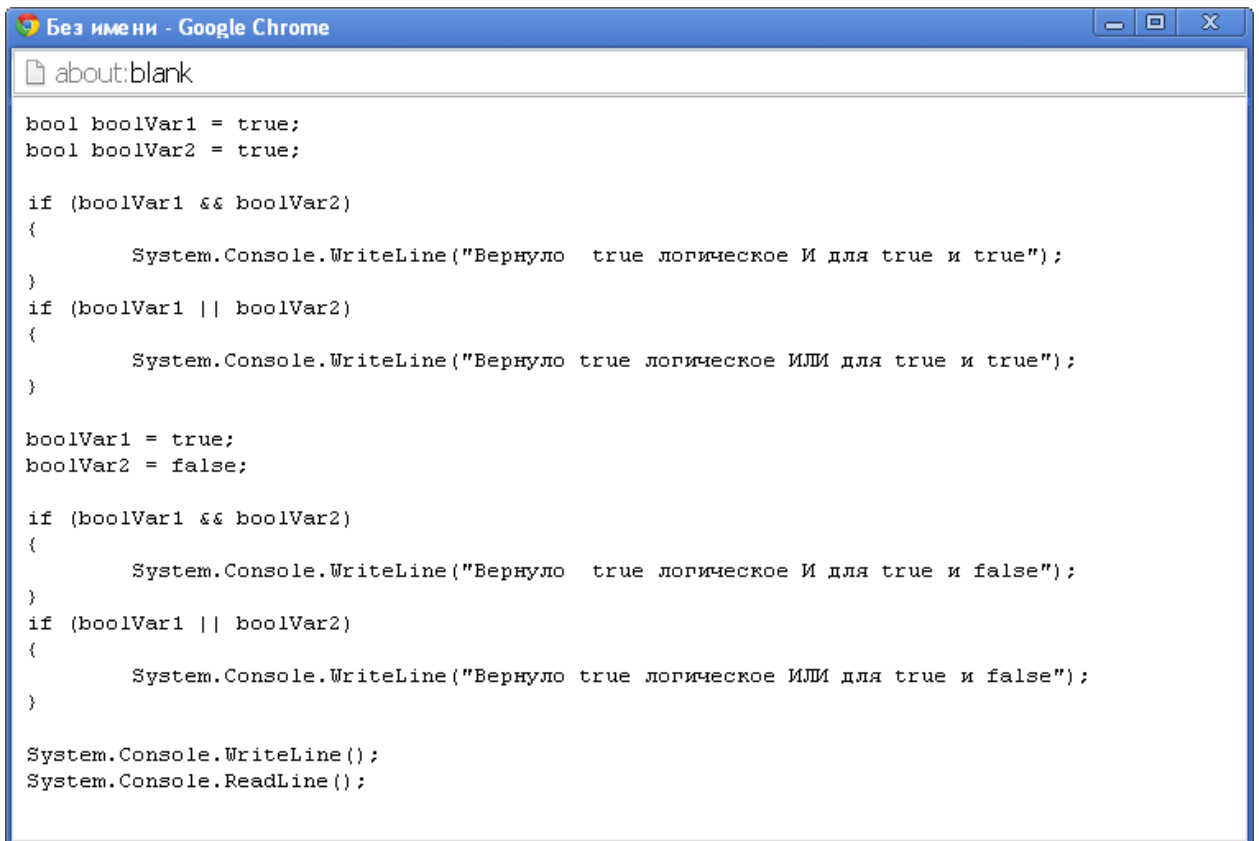
System.Console.WriteLine();

System.Console.ReadLine();
```

Сделать поведение программы еще более сложным нам помогут специальные логические операторы, которые сравнивают две логические величины (каждая из которых может быть либо истиной, либо ложью)

- логическое И - [оператор &&](#) - возвращает истину только в том случае, если и справа и слева от него будет истина, во всех остальных случаях будет ложь.

- логическое ИЛИ - [оператор ||](#) - возвращает истину, если хоть одна из двух величин истинна. Ложь он вернет, только если обе логические величины ложны.



```
bool boolVar1 = true;
bool boolVar2 = true;

if (boolVar1 && boolVar2)
{
    System.Console.WriteLine("Вернуло true логическое И для true и true");
}
if (boolVar1 || boolVar2)
{
    System.Console.WriteLine("Вернуло true логическое ИЛИ для true и true");
}

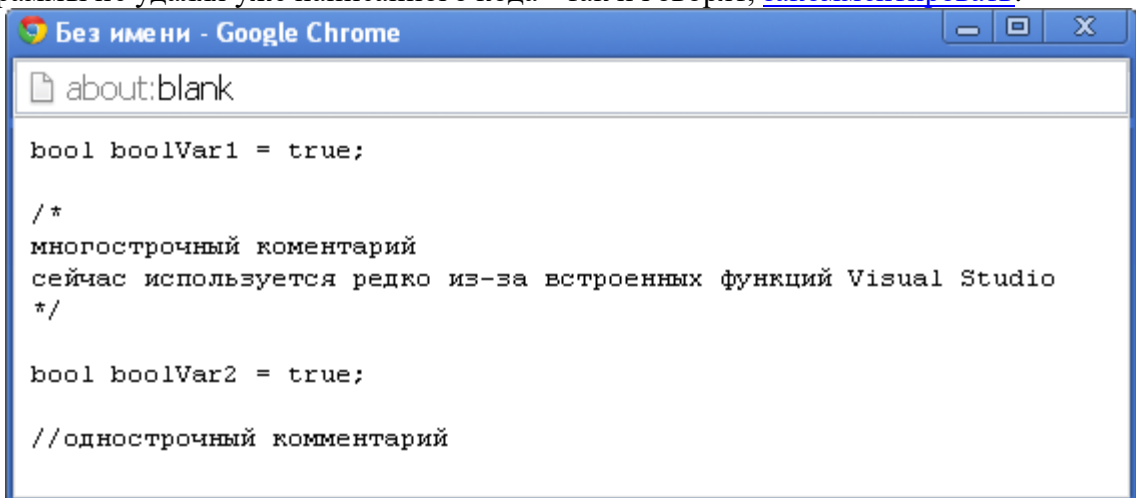
boolVar1 = true;
boolVar2 = false;

if (boolVar1 && boolVar2)
{
    System.Console.WriteLine("Вернуло true логическое И для true и false");
}
if (boolVar1 || boolVar2)
{
    System.Console.WriteLine("Вернуло true логическое ИЛИ для true и false");
}

System.Console.WriteLine();
System.Console.ReadLine();
```

Задание 5. Комментарии

До этого момента наши тестовые программки были просты. Дальше они постепенно начнут усложняться. Настоящим программы бывают сложными. Очень сложным. На самом деле самая сложная часть работы программиста - это не написать новую программу, а понять уже написанную, найти в ней ошибки, исправить их или что-то поменять не сделав новых. Понимание программы очень облегчают комментарии - заметки прямо в коде, которые игнорирует компьютер. Они только для высшего разума - людей.)) Кроме того их часто используют для того, чтобы временно отключить часть программы не удаляя уже написанного кода - так и говорят, [закомментировать](#).



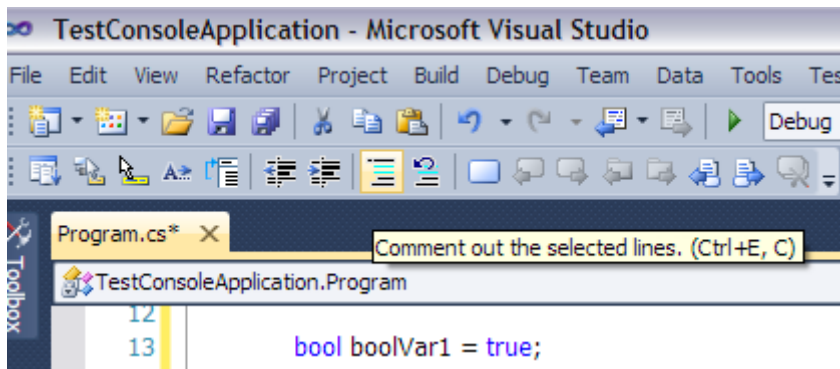
```
bool boolVar1 = true;

/*
многострочный комментарий
сейчас используется редко из-за встроенных функций Visual Studio
*/

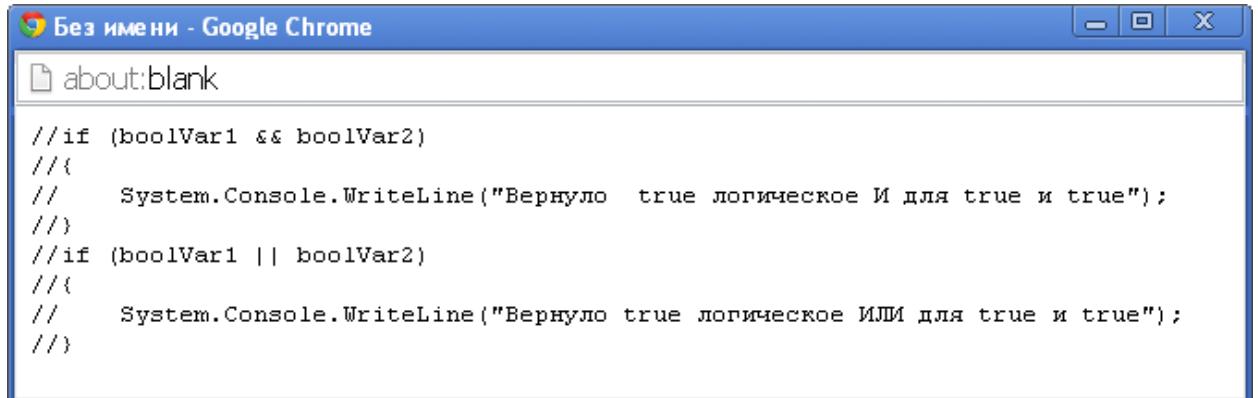
bool boolVar2 = true;

//однострочный комментарий
```

На практике многострочные комментарии в Visual Studio чаще всего ставятся с использованием специальной функции - у нее есть и кнопка и собственная горячая клавиша



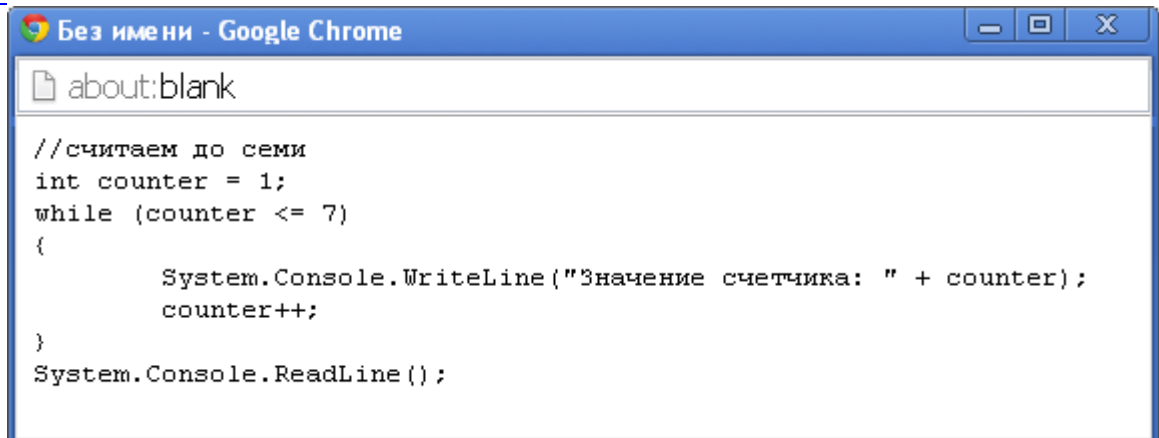
Получаем вот такой код, который можно раскомментировать с помощью соседней кнопки



Задание 6. Циклы

Настоящие роботы славятся своей способностью повторять любые действия неограниченное количество раз не зная усталости и сомнений! Делают они это с помощью циклов - специальных конструкция для программирования повторяющихся действий.

Самый простой цикл просто заставляет компьютер повторять какое-то действие, пока не выполнится какое-то логическое (см. выше!) условие. В C# такой цикл называется [while](#)

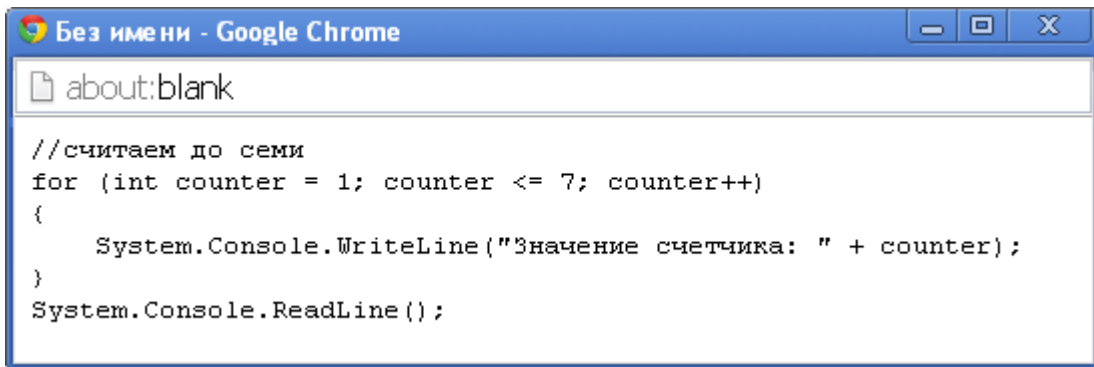


Обратите внимание: если условие не выполнится никогда, компьютер будет выполнять заданное действие вечно - это называется заикливанием.

Можете проверить это утверждение закомментировав строку counter++; - программа будет работать бесконечно, так как counter никогда не будет равен 7. Придется прекратить ее работу вручную, закрыв окно.

Делать что-то заданное количество раз приходится так часто, что для этого случая придумали специальный цикл [for](#).

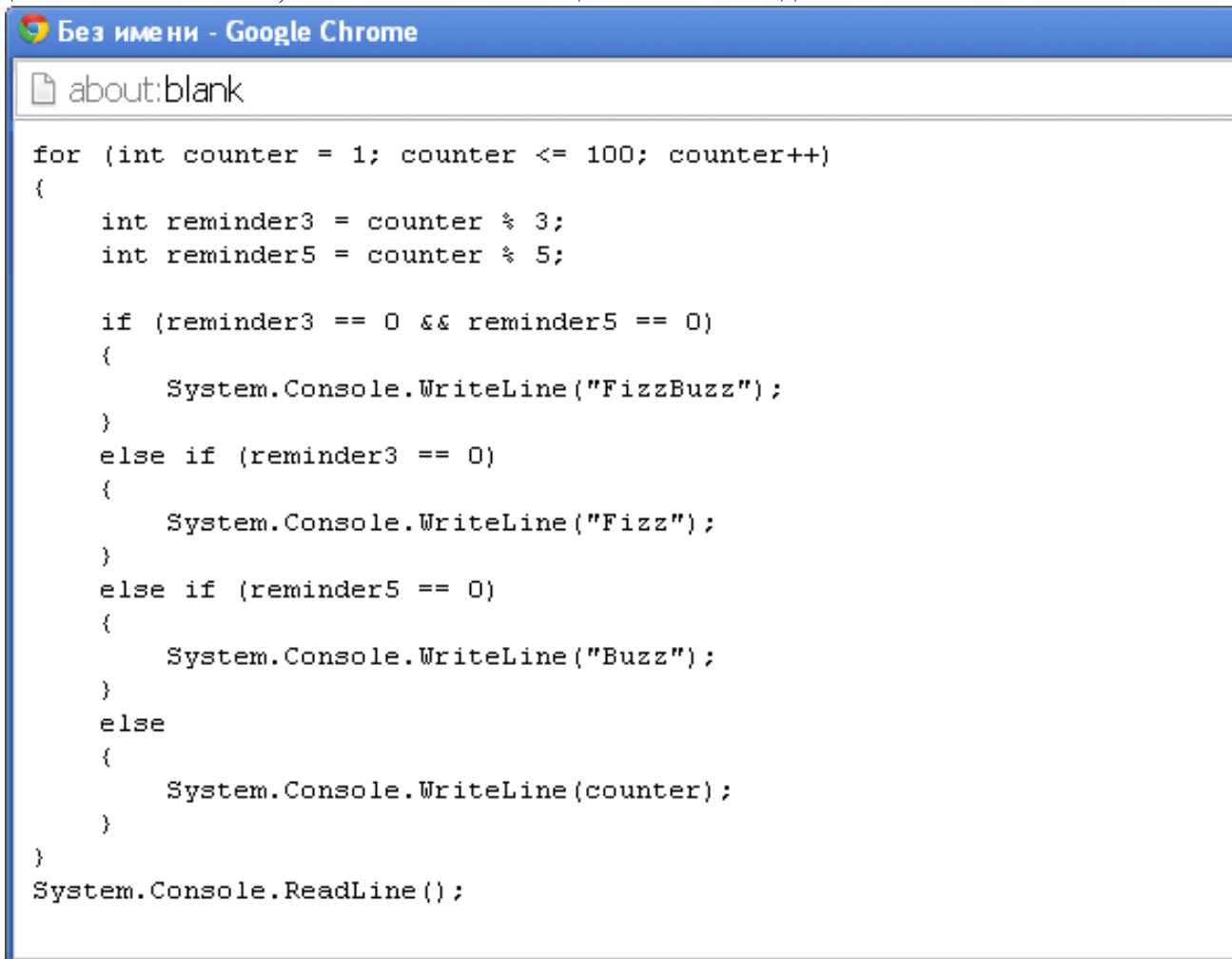
С его помощью даже самый простой цикл из предыдущего примера станет еще проще



```
//считаем до семи
for (int counter = 1; counter <= 7; counter++)
{
    System.Console.WriteLine("Значение счетчика: " + counter);
}
System.Console.ReadLine();
```

В заменитой статье [Почему программисты не умеют программировать](#) автор приводит следующие данные - большинство выпускников программистских вузов не могут написать очень простую программу - которая выведет на на экран числа от 1 до 100, вместо кратных 3 выведет Fizz, вместо кратных 5 выведет Buzz, вместо кратных и трем и пяти FizzBuzz ([это в Америке такая обучающая игра для детей](#)). На самом деле единственная сложность в такой задаче - вспомнить [школьное определение кратности](#). Таким образом число кратно 3 если остаток от целочисленного (без учета дробной части) деления этого числа на 3 равен нулю. Осталось выяснить, как вычислить остаток от целочисленного деления. Это делает [оператор %](#).

В итоге получаем совсем простую программу, то ли в Америке так плохо учат, то ли автор статьи преувеличивает, то ли просто большинству программистов никогда не понадобилось знание того, что такое остаток от целочисленного деления.

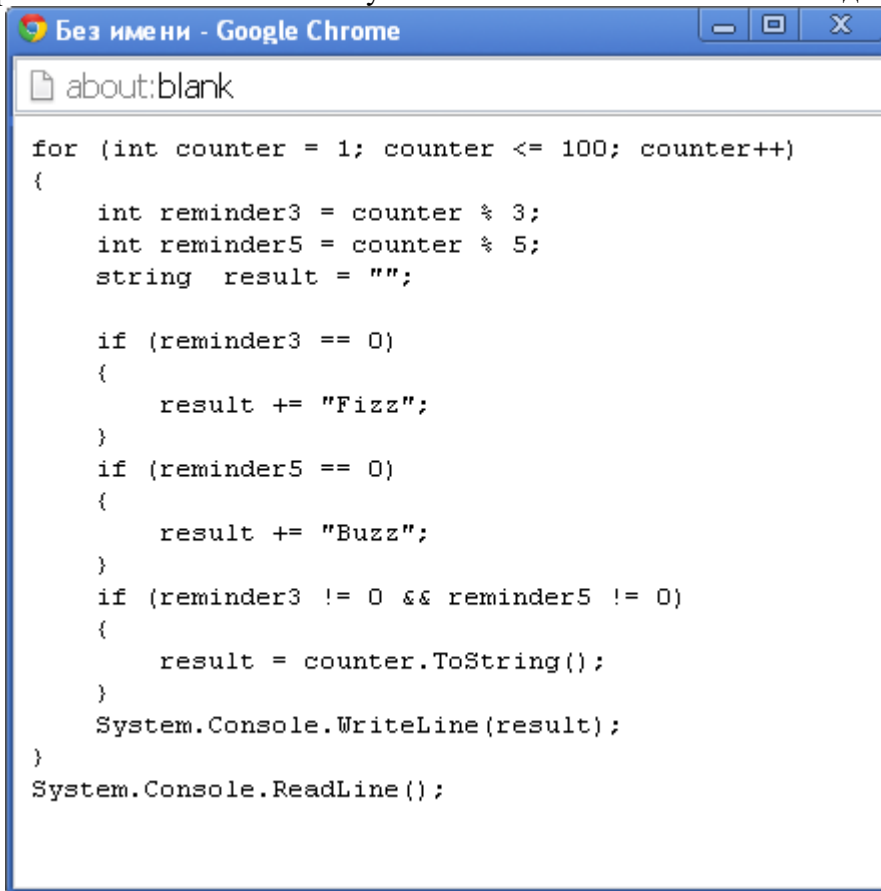


```
for (int counter = 1; counter <= 100; counter++)
{
    int remainder3 = counter % 3;
    int remainder5 = counter % 5;

    if (remainder3 == 0 && remainder5 == 0)
    {
        System.Console.WriteLine("FizzBuzz");
    }
    else if (remainder3 == 0)
    {
        System.Console.WriteLine("Fizz");
    }
    else if (remainder5 == 0)
    {
        System.Console.WriteLine("Buzz");
    }
    else
    {
        System.Console.WriteLine(counter);
    }
}
System.Console.ReadLine();
```

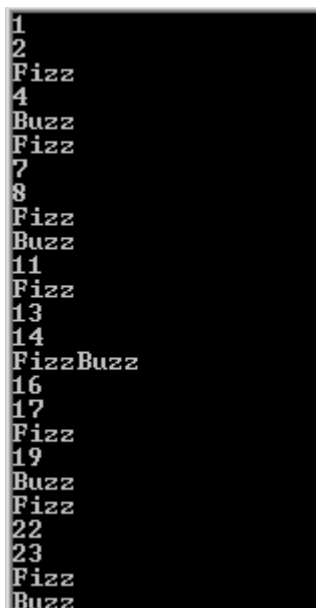
Опытный программист может заметить, что одно из условий if лишнее и код можно сделать более коротким с использованием простого трюка - если число делится и на 3 и на

5, то можно просто выводить сначала Fizz потом Buzz, автоматически склеивая их в одну строку и не проверяя отдельно на предмет кратности и 3 и 5. Так как все в этом мире имеет свою цену, то более короткий код также будет менее понятным. В данном примере различие невелико, но в промышленном программировании со сложными и большими программами в большинстве случаев важна именно понятность кода.



```
Без имени - Google Chrome
about:blank
for (int counter = 1; counter <= 100; counter++)
{
    int remainder3 = counter % 3;
    int remainder5 = counter % 5;
    string result = "";

    if (remainder3 == 0)
    {
        result += "Fizz";
    }
    if (remainder5 == 0)
    {
        result += "Buzz";
    }
    if (remainder3 != 0 && remainder5 != 0)
    {
        result = counter.ToString();
    }
    System.Console.WriteLine(result);
}
System.Console.ReadLine();
```



```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
Fizz
22
23
Fizz
Buzz
```

Задание 7.

Написать C# программу, реализующую функцию согласно варианту задания. Исходные данные вводятся с клавиатуры. варианты выберите по списку в группе. (3 задачи каждому студенту)

1. Реализовать функцию вычисления суммы двух целых чисел.
2. Реализовать функцию вычисления разности двух целых чисел.
3. Реализовать функцию вычисления произведения двух целых чисел.
4. Реализовать функцию вычисления частного двух целых чисел.
5. Реализовать функцию вычисления суммы двух вещественных чисел.
6. Реализовать функцию вычисления разности двух вещественных чисел.
7. Реализовать функцию вычисления произведения двух вещественных чисел.
8. Реализовать функцию вычисления частного двух вещественных чисел.
9. Реализовать функцию возведения целого числа в квадрат.
10. Реализовать функцию возведения в квадрат суммы двух целых чисел.
11. Реализовать функцию возведения в квадрат разности двух целых чисел.
12. Реализовать функцию возведения в квадрат произведения двух целых чисел.
13. Реализовать функцию возведения в квадрат частного двух целых чисел.
14. Реализовать функцию возведения в квадрат суммы двух вещественных чисел.
15. Реализовать функцию возведения в квадрат разности двух вещественных чисел.
16. Реализовать функцию возведения в квадрат произведения двух вещественных чисел.
17. Реализовать функцию возведения в квадрат частного двух вещественных чисел.
18. Реализовать функцию возведения в куб целого числа.
19. Реализовать функцию возведения в куб суммы двух целых чисел.
20. Реализовать функцию возведения в куб разности двух целых чисел.
21. Реализовать функцию возведения в куб произведения двух целых чисел.
22. Реализовать функцию возведения в куб частного двух целых чисел.
23. Реализовать функцию возведения в куб суммы двух вещественных чисел.
24. Реализовать функцию возведения в куб разности двух вещественных чисел.
25. Реализовать функцию возведения в куб произведения двух вещественных чисел.
26. Реализовать функцию возведения в куб частного двух вещественных чисел.
27. Реализовать функцию вычисления частного двух вещественных чисел.
28. Реализовать функцию возведения целого числа в квадрат.
29. Реализовать функцию возведения в квадрат разности двух целых чисел.
30. Реализовать функцию возведения в квадрат произведения двух целых чисел.

Указания по технике безопасности

При выполнении практической работы **запрещается:**

1. Самостоятельно производить ремонт персонального компьютера, а так же, установку и удаление имеющегося программного обеспечения;
2. Нарушать общепринятые правила техники безопасности при работе с электрооборудованием, в частности, касаться электрических розеток металлическими предметами и т.д.;

3. Принимать пищу, напитки и сорить на рабочем месте пользователя персонального компьютера.

В случае неисправности персонального компьютера необходимо **немедленно** сообщить об этом обслуживающему персоналу лаборатории (системному администратору, оператору).

Содержание отчета и его форма

Отчет по практической работе оформляется в виде текстового документа и должен включать:

1. Название практической работы.
2. Цель практической работы.
3. Формулировку индивидуального задания и результат его выполнения.
4. Краткие выводы по результатам выполнения практической работы.

Защита практической работы

Оформленный, в соответствии с требованиями ГОСТов, отчет представляется студентом преподавателю для проверки и последующей защиты. Защита отчета по практической работе производится студентом только индивидуально.

В ходе защите практической работы студент отвечает на вопросы преподавателя (поясняет методику выполнения индивидуального задания, отвечает на контрольные вопросы и т.д.).

Отчет, оформленный с отступлениями от требований ГОСТов, небрежно и неаккуратно к защите не принимается.

Практические работы № 9-10. Создание личной страницы студента с помощью сервисов Google

Цель и содержание: изучить основные возможности сервиса Google «Сайты». Разработать личную страницу, используя сервис Google «Сайты».

Аппаратура и программное обеспечение.

Аппаратура. Для выполнения практической работы необходимы:

- персональный компьютер
- рабочие станции: монитор + клавиатура + мышь.

Программное обеспечение. Для выполнения лабораторной работы необходима операционная система WINDOWS 7/8/10 и подключение к глобальной сети Интернет.

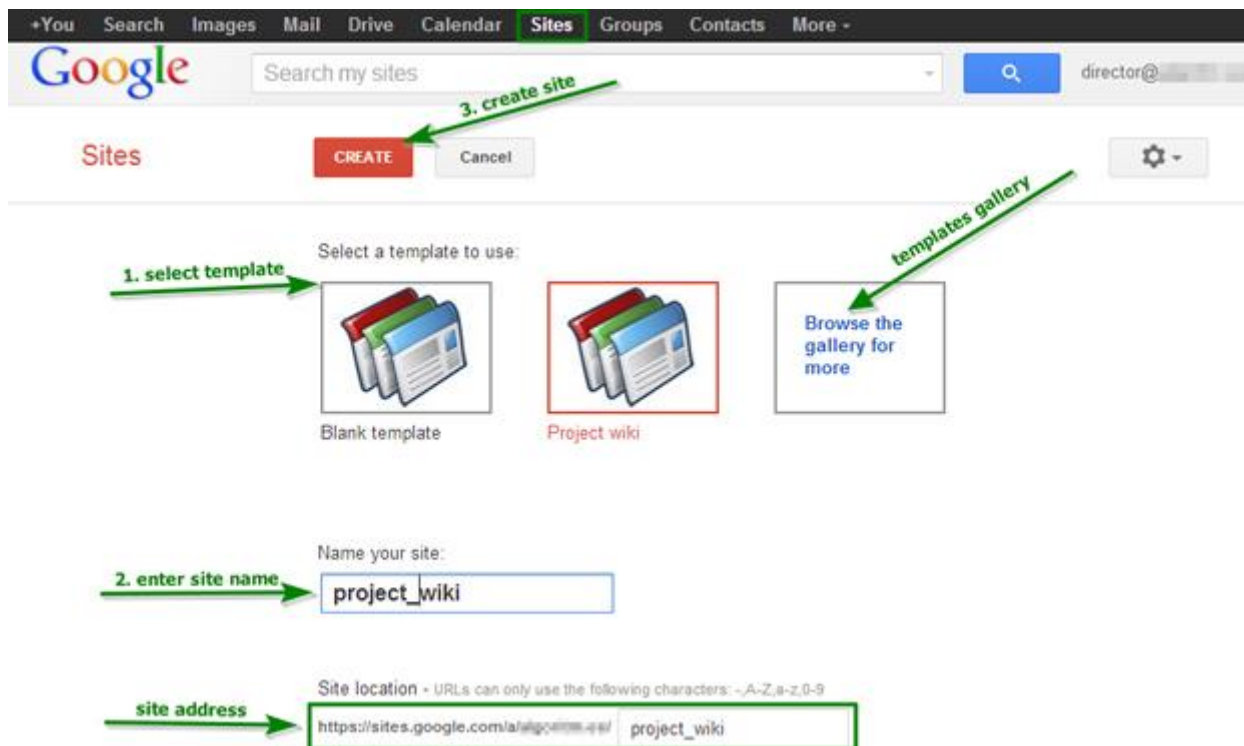
Методика и порядок выполнения работы

Теоретический материал

Сервис "Сайты". Этот сервис предназначен для создания мини-сайтов для совместной работы и централизованного хранения связанных между собой документов. В качестве примеров использования сервиса "Сайты" могут служить, например, *сайт* отдельного факультета/подразделения учебного заведения или *wiki* приоритетного научного проекта. Для создания собственных сайтов образовательному учреждению выделяется 10 ГБ дискового пространства в облаке, плюс 500 МБ для хранения файлов каждого сотрудника.

Чтобы создать новый *сайт*, необходимо перейти в сервис "Сайты" и выбрать "Create". Проще всего создать *сайт* на основе готового шаблона, который можно выбрать из внушительного списка, представленного в галерее шаблонов.

На [рис. 2.44](#) с помощью стрелок подробно показан процесс создания нового сайта из шаблона. Порядок выполнения действия показан нумерацией соответствующих стрелок.



[увеличить](#)

[изображение](#)

Рис. 2.44. Создание сайта на основе шаблона

После выбора шаблона и ввода названия сайта необходимо нажать "Create" для запуска процедуры генерации сайта. Генерация сайта на основе шаблона может занять одну-две минуты. Как только сайт будет создан, появится возможность редактировать существующие страницы и добавлять новые (в качестве подробного пример см. [рис. 2.45](#)).



[увеличить](#)

[изображение](#)

Рис. 2.45. Интерфейс редактирования веб-страницы

Процесс редактирования веб-страниц не вызывает особых сложностей, поскольку используется тот же самый интерфейс, что и при редактировании документов в сервисе "Documents". Предоставление общего доступа к сайту осуществляется так же, как и

предоставление общего доступа к документам – необходимо выбрать пункт "Share" интерфейса, затем настроить видимость и права доступа к сайту (см. [рис. 2.46](#)).

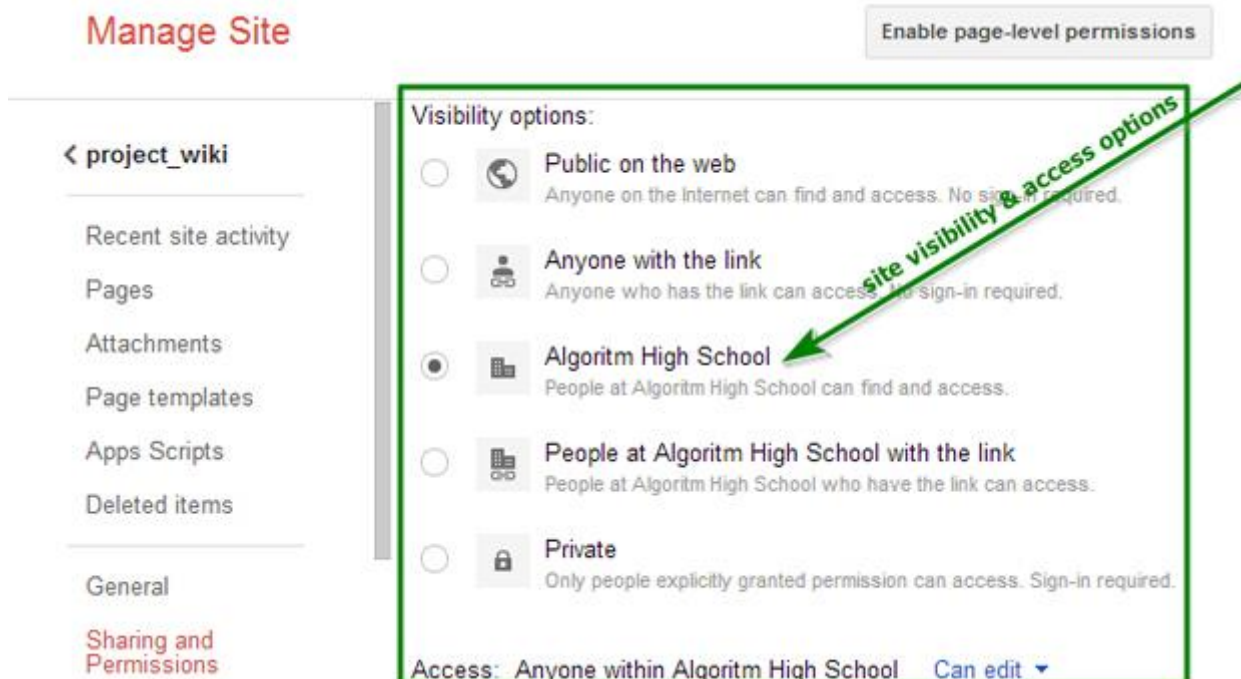


Рис. 2.46. Управление настройками сайта

Сервис "Сайты" позволяет быстро и просто создавать локальные сайты учебного заведения. Этот инструмент позволяет преподавателям сконцентрироваться на создании качественных учебных материалов, сокращая при этом время на их оформление и публикацию, что позволяет сделать процесс обучения более продуктивным.

Служба поиска. В настоящее время компания Google является мировым лидером *по* предоставлению поисковых услуг в сети *Интернет*. Поэтому во всех приложениях Google Apps присутствует инструмент поиска, будь то сервис электронной почты, календарь или документы. *Поиск* осуществляется не только *по* имени файла, электронного письма или события календаря, но и *по* содержанию этих объектов. Даже если ученик или преподаватель забыл, как называется тот или иной документ, найти его можно *по* одному или нескольким словам, которые содержатся в этом документе.

Как и ранее, пример того как выглядит *интерфейс GAfE*, использующийся при работе службы поиска, представлен на [рис. 2.47](#). Активные элементы, использующиеся при поиске, а также результаты поиска отмечены стрелками.

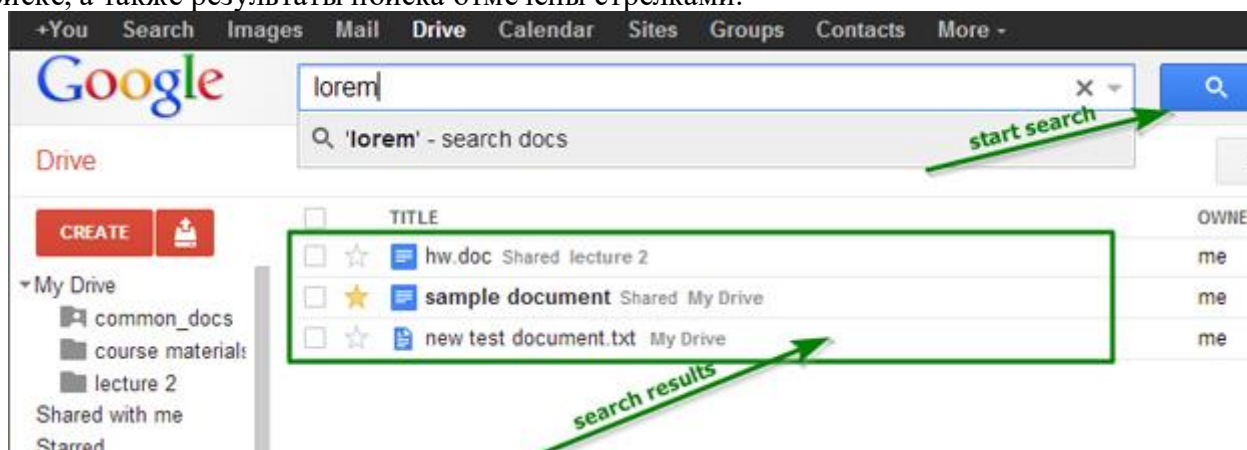


Рис. 2.47. Поиск документов по содержимому

Сложностей с поиском информации не возникает, поскольку в Google Apps for Education используется тот же *интерфейс*, что и на сайте поисковой машины Google. Стоит обратить особое внимание на то, что *поиск* осуществляется только в рамках отдельной службы. Таким образом, чтобы найти документ, необходимо перейти в сервис "Drive", а для поиска электронного письма или сообщения, соответственно, - в сервис "Mail". Поиск по всем объектам учебного заведения возможен в сервисах "Drive", "Calendar", "Sites", который можно осуществить в режиме расширенного поиска.

Выполнить задание

1. Зайти под своим профилем в сервисы Google. Выбрать сервис «Сайты»
2. Создать 5 страниц с простым шаблоном: Обо мне, Моя группа, Мои увлечения, Мои друзья, Моя учеба.
3. Посмотреть возможности изменения формата страниц
4. Посмотреть возможности изменения шаблона страниц
5. Отправить преподавателю по адресу zamira.ncfu@gmail.com ссылку на созданную страницу студента.

Указания по технике безопасности

При выполнении практической работы **запрещается**:

1. Самостоятельно производить ремонт персонального компьютера, а так же, установку и удаление имеющегося программного обеспечения;
2. Нарушать общепринятые правила техники безопасности при работе с электрооборудованием, в частности, касаться электрических розеток металлическими предметами и т.д.;
3. Принимать пищу, напитки и сорить на рабочем месте пользователя персонального компьютера.

В случае неисправности персонального компьютера необходимо **немедленно** сообщить об этом обслуживающему персоналу лаборатории (системному администратору, оператору).

Содержание отчета и его форма

Отчет по лабораторной работе оформляется в виде текстового документа и должен включать:

1. Название лабораторной работы.
2. Цель лабораторной работы.
3. Формулировку индивидуального задания и результат его выполнения.
4. Краткие выводы по результатам выполнения практической работы.

Защита практической работы

Оформленный, в соответствии с требованиями ГОСТов, отчет представляется студентом преподавателю для проверки и последующей

защиты. Защита отчета по практической работе производится студентом только индивидуально.

В ходе защите лабораторной работы студент отвечает на вопросы преподавателя (поясняет методику выполнения индивидуального задания, отвечает на контрольные вопросы и т.д.).

Отчет, оформленный с отступлениями от требований ГОСТов, небрежно и неаккуратно к защите не принимается.

Практические работы № 11 – 12. Язык html. Разработка стиля и архитектуры Web-сайта

Цель и содержание: Разработать собственный web-дизайн, изучить основные теги языка html, создать первые html-страницы.

Аппаратура и программное обеспечение.

Аппаратура. Для выполнения практической работы необходимы:

- персональный компьютер

- рабочие станции: монитор + клавиатура + мышь.

Программное обеспечение. Для выполнения лабораторной работы необходима операционная система WINDOWS 7/8/10 и подключение к глобальной сети Интернет.

Методика и порядок выполнения работы

Задание 1. Разобраться с понятиями дизайн и web-дизайн.

Ниже приводится подборка материала, в той или иной мере отвечающей на поставленный вопрос. Прочтите и составьте свое представление по следующим вопросам.

Что		такое	дизайн
Что	же	такое	web-дизайн?
Как	делать	сайты	качественнее
Вебмастеринг:		четыре	шага
Основные	ошибки	начинающих	дизайнеров
Маркетинговые	принципы	построения	виртуальных страниц
В	очередь	за	астроблюбой
Немного об index.html			

Задание 2. Язык разметки гипертекста HTML.

В практическом руководстве по HTML даются основные характеристики этого языка, с примерами использования. При работе часто будут возникать различные вопросы типа: "Как это сделать, чтобы было красиво?". На многие подобные вопросы можно найти ответы в FAQ of RU.HTML.CHAINIK. При работе с гипертекстовыми документами, значительную помощь вам может оказать справочник по тегам.

Для работы с гипертекстом необходимо использовать HtmlPad, Far или Блокнот.

Ознакомившись с основными требованиями, тегами и способами форматирования создайте следующие страницы: **Обо мне** (автобиография, т.е. где родились, где учились, что умеете лучше всего делать и т.п.), **Мои друзья** (сведения о ваших школьных и институтских друзьях: ФИО, чем увлекаются, сколько лет знакомы и т.д.), **Мои увлечения** (то, чем вы любите заниматься в свободное от учебы время, какие фильмы, музыка и произведения вам нравятся, описав при этом, почему вы сделали такой выбор и др.), **Я и моя группа** (написать свои отзывы о каждом из одноклассников и разместив рядом их

фотографию). Данные сведения будут записаны вам и вашим одногруппникам на диски, так что постарайтесь описать все это достойным образом! Поместите созданные файлы в каталог **about**, и не забывайте - **ВСЕ ФАЙЛЫ ДОЛЖНЫ ИМЕТЬ ЛАТИНСКИЕ НАЗВАНИЯ!** Например: *about.htm*, *friends.htm*, *hobby.htm* и *groop.htm*. Фотографии одногруппников должны находиться в том же каталоге.

Если при создании страниц возникают трудности при форматировании текста воспользуйтесь "*Руководством по HTML*", упомянутым выше, или, данным примером. При необходимости вставить в документ символ воспользуйтесь таблицей спец. символов.

Задание 3. Выбор цветовой палитры сайта.

Цветовая палитра является одним из самых важных элементов оформления Web сайта. Познакомьтесь с основными характеристиками цвета: {яркость, оттенок, насыщенность} {яркость} {насыщенность} {оттенок}.

Для подбора цвета своей страницы вы можете воспользоваться редактором цвета.

Используя таблицы цветов, выберите палитру из 18 рядом лежащих цветов, которые вы будете использовать для своего сайта. Оформите отдельную страничку "*palitra.htm*" (*Набор палитр*) с этим набором и разместите ее в разделе "**Учебная работа**". Для помощи можете воспользоваться **Набором палитр для WEB-дизайна**. Выбрав подходящее сочетание цветов можете сделать скриншот и, обратив изображение при помощи Photoshop, подвязать его к своей странице. Создайте в Photoshop заголовок для данной страницы размером 800x100 и поместите его в самом начале. Его стиль должен соответствовать выбранному стилю сайта.

Задание 4. Создание гиперссылок.

В результате проделанной работы у вас должно получиться 6 html-страниц, которые необходимо объединить гиперссылками.

Указания по технике безопасности

При выполнении лабораторной работы **запрещается**:

1. Самостоятельно производить ремонт персонального компьютера, а так же, установку и удаление имеющегося программного обеспечения;
2. Нарушать общепринятые правила техники безопасности при работе с электрооборудованием, в частности, касаться электрических розеток металлическими предметами и т.д.;
3. Принимать пищу, напитки и сорить на рабочем месте пользователя персонального компьютера.

В случае неисправности персонального компьютера необходимо **немедленно** сообщить об этом обслуживающему персоналу лаборатории (системному администратору, оператору).

Содержание отчета и его форма

Отчет по лабораторной работе оформляется в виде текстового документа и должен включать:

1. Название практической работы.
2. Цель практической работы.
3. Формулировку индивидуального задания и результат его выполнения.
4. Краткие выводы по результатам выполнения практической работы.

Защита лабораторной работы

Оформленный, в соответствии с требованиями ГОСТов, отчет представляется студентом преподавателю для проверки и последующей защиты. Защита отчета по практической работе производится студентом только индивидуально.

В ходе защите лабораторной работы студент отвечает на вопросы преподавателя (поясняет методику выполнения индивидуального задания, отвечает на контрольные вопросы и т.д.).

Отчет, оформленный с отступлениями от требований ГОСТов, небрежно и неаккуратно к защите не принимается.

Практические работы № 13 – 15. Создание раздела "Библиотека". Разработка каскадных таблиц стилей (CSS)

Цель и содержание работы: Если Вы решили создать HTML-документ с разветвленной структурой, постарайтесь оформить все его страницы одинаково, оперируя в рамках одного стиля. Применяйте повторяющиеся управляющие элементы, одинаковые интервалы и подобные шрифты. Это относится и к регулярному обновлению Web-страниц. Чтобы добиться стилевого единства, можно создать ряд шаблонов, которые подлежат заполнению в текстовом редакторе. Временные затраты на разработку шаблонов быстро окупятся при больших объемах документов и частом обновлении содержимого.

Аппаратура и программное обеспечение.

Аппаратура. Для выполнения работы необходимы:

- персональный компьютер

- рабочие станции: монитор + клавиатура + мышь.

Программное обеспечение. Для выполнения лабораторной работы необходима операционная система WINDOWS 7/8/10 и подключение к глобальной сети Интернет.

Методика и порядок выполнения работы

Задание 1. Ознакомьтесь со способами создания и применения каскадных таблиц стилей.

Спецификация CSS (Cascading Style Sheets) позволяет остаться в рамках декларативного характера разметки страницы и дает полный контроль над формой представления элементов HTML-разметки. Изучите назначение, способы применения и синтаксис для CSS.

Создайте CSS- файл style.css, внутри которого будет задан **ВАШ** стиль для страниц, входящих в состав раздела "Библиотека". А именно цвет и стиль шрифта, цвет и стиль ссылок, а также стиль объектов и текста, помещенных между тегами, которые вы использовали при создании своих страниц (например цвет и положение списков, положение рисунков и т.д.).

В корневом каталоге (public_html) создайте каталог **CSS**, в который поместите CSS - файл, чтобы им можно было пользоваться в дальнейшем.

Задание 2. Создание раздела "Библиотека".

Работа по стилизованию оформления страниц весьма трудоемкая и требует знаний различных нюансов. Познакомьтесь с некоторыми советами специалистов.

- Подбор шрифта и верстка текста.
- Правильные таблицы стилей
- Доступ к атрибутам стиля в IE и NN

Создайте страницу "**biblio.htm**" (**Библиотека**) с использованием ссылки на внешнее описание применяемых стилей (css) через элемент *LINK*.

На странице создать меню, состоящее из следующих пунктов:

- Научная литература
- Фантастика
- Юмор
- Стихи
- Статьи

Ниже меню дайте описание вышеперечисленных разделов, включив в него перечисление произведений, вошедших в данный раздел, их автора и краткую характеристику каждого из произведений. Произведения смотрите ниже.

Задание 3. Оформление раздела Библиотека.

Теперь вам необходимо оформить "Библиотеку". Для начала, вам необходимо отредактировать текст в MS Word или в OpenOffice.org Writer, точнее - создать страницы своей библиотеки и проверить при помощи средств орфографии и пунктуации правильность написания создаваемых документов. В каждом разделе библиотеки будет по 2-3 произведения. Пока что все их можно создавать в одном документе, а затем, когда вы будете их перегонять в формат html, их необходимо будет расположить на разных страницах.

Каждый должен выбрать свой вариант задания.

- Научная литература
- Фантастика
- Юмор
- Стихи
- Статьи

Пример содержания файла CSS

```
<STYLE type="text/css">
<!--
```

```
h2 { text-align:center; font-family:Times New Roman;} - текст, заключенный между тегом <h2> и </h2> выравнивается по центру, шрифт Times New Roman
```

```
p {padding-left:40px;padding-right:42px; text-align:justify; border-width:1px; font-size:13pt;} - текст, заключенный между тегом <p> и </p> имеет отступ слева 40px, справа - 42, текст выравнивается по ширине, размер текста 13 пунктов.
```

```
pre {padding-left:40px;padding-right:42px; text-align:left; font-size:10pt; font-family:Courier New;}
```

```
font {text-align:left; border-width:1px; font-size:11pt; font-family:Courier New;}
```

```
hr {width:93%; height:1px;} - все линии, вставленные в документ, будут иметь ширину 93% от размера окна и высоту 1 пиксел.
```

```
h3 {padding-left:30px;padding-right:32px; text-align:justify;}
```

```
ol {padding-left:30px;padding-right:32px; text-align:justify;}
```

```
ul {padding-left:30px;padding-right:32px; text-align:justify;}
```

```
A {TEXT-DECORATION: none;} - говорит о том, ссылки не будут подчеркнуты
```

```
A:link {color:#006699;} - цвет ссылки
```

```
A:visited {color:#3399FF;} - цвет ссылок, на которые вы уже заходили
```

```
A:active {color: #3399FF ;} - цвет активной ссылки, т.е. в момент нажатия
```

```
A:hover {color:#443323; TEXT-DECORATION: none;} - цвет ссылки при наведении указателя мыши
```

```
--!>
```

```
</STYLE>
```

Указания по технике безопасности

При выполнении практической работы **запрещается**:

1. Самостоятельно производить ремонт персонального компьютера, а так же, установку и удаление имеющегося программного обеспечения;
2. Нарушать общепринятые правила техники безопасности при работе с электрооборудованием, в частности, касаться электрических розеток металлическими предметами и т.д.;
3. Принимать пищу, напитки и сорить на рабочем месте пользователя персонального компьютера.

В случае неисправности персонального компьютера необходимо **немедленно** сообщить об этом обслуживающему персоналу лаборатории (системному администратору, оператору).

Содержание отчета и его форма

Отчет по практической работе оформляется в виде текстового документа и должен включать:

1. Название практической работы.
2. Цель лабораторной работы.
3. Формулировку индивидуального задания и результат его выполнения.
4. Краткие выводы по результатам выполнения практической работы.

Защита практической работы

Оформленный, в соответствии с требованиями ГОСТов, отчет представляется студентом преподавателю для проверки и последующей защиты. Защита отчета по практической работе производится студентом только индивидуально.

В ходе защите лабораторной работы студент отвечает на вопросы преподавателя (поясняет методику выполнения индивидуального задания, отвечает на контрольные вопросы и т.д.).

Отчет, оформленный с отступлениями от требований ГОСТов, небрежно и неаккуратно к защите не принимается.

Практические работы № 16 – 17. Использование таблиц

Цель и содержание: Таблицы нужны для представления информации в табличном виде. Есть, однако, и менее очевидные способы применения таблиц. До настоящего времени мы имели дело с документами, в которых существовал только один "поток" текста. На практике иногда очень хочется расположить текст в несколько колонок. Таблицы могут в этом помочь. Кроме того, таблица, состоящая из одной ячейки, может очень эффектно выделить фрагмент текста, на который Вы хотите обратить внимание читателя вашей страницы.

Аппаратура и программное обеспечение.

Аппаратура. Для выполнения практической работы необходимы:

- персональный компьютер
- рабочие станции: монитор + клавиатура + мышь.

Программное обеспечение. Для выполнения лабораторной работы необходима операционная система WINDOWS 7/8/10 и подключение к глобальной сети Интернет.

Методика и порядок выполнения работы

Задание 1. Как устроены таблицы.

Самый простой механизм размещения материала сайта - это таблицы. Это самый распространенный и почти единственный инструмент позиционирования объектов веб-страницы. Обычно таблицу и используют исключительно для этих целей. Чаще всего с помощью таблиц составляют сложные композиции из отдельных рисунков.

Очень внимательно разберитесь с методами построения страницы. Для работы с исходным текстом этой страницы можно сохранить страницу в HTML-виде браузером.

Задание 2. Таблицы в примерах

Чтобы разобраться с технологией построения таблиц можете воспользоваться примерами построения таблиц различной конфигурации. При создании таблиц можно устанавливать распорки, удерживающие ее размеры.

Таблицы в примерах

Этот документ содержит примеры таблиц. Он охватывает большинство новых тэгов в таблицах. Впрочем он не обязательно показывает некоторые действительно творческие возможности, доступные в таблицах.

ОБЫЧНАЯ ТАБЛИЦА 3X2


```
<TABLE BORDER>  
<TR>  
<TD>A</TD> <TD>B</TD> <TD>C</TD>
```

```

</TR>
<TR>
  <TD>D</TD> <TD>E</TD> <TD>F</TD>
</TR>
</TABLE>

```

ДВА ПРИМЕРА С ROWSPAN

Ite m 1	Ite m 2	Ite m 3
Ite m 4		Ite m 5

```

<TABLE BORDER>
  <TR>
    <TD>Item 1</TD>
    <TD ROWSPAN=2>Item 2</TD>
    <TD>Item 3</TD>
  </TR>
  <TR>
    <TD>Item 4</TD> <TD>Item 5</TD>
  </TR>
</TABLE>

```

Ite m 1	Ite m 2	Ite m 3	Ite m 4
	Ite m 5	Ite m 6	Ite m 7

```

<TABLE BORDER>
  <TR>
    <TD ROWSPAN=2>Item 1</TD>
    <TD>Item 2</TD> <TD>Item 3</TD> <TD>Item 4</TD>
  </TR>
  <TR>
    <TD>Item 5</TD> <TD>Item 6</TD> <TD>Item 7</TD>
  </TR>
</TABLE>

```

ПРИМЕР С COLSPAN

Ite m 1	Item 2	
Ite m 3	Ite m 4	Ite m 5

```

<TABLE BORDER>
  <TR>
    <TD>Item 1</TD>
    <TD COLSPAN=2>Item 2</TD>
  </TR>
  <TR>
    <TD>Item 3</TD> <TD>Item 4</TD> <TD>Item 5</TD>
  </TR>

```

```
</TR>
</TABLE>
```

ДЕМОНСТРАЦИЯ ЗАГОЛОВКОВ (<TH>)

He ad1	He ad2	He ad3
A	B	C
D	E	F

```
<TABLE BORDER>
```

```
<TR>
```

```
<TH>Head1</TH> <TH>Head2</TH> <TH>Head3</TH>
```

```
</TR>
```

```
<TR>
```

```
<TD>A</TD> <TD>B</TD> <TD>C</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD>D</TD> <TD>E</TD> <TD>F</TD>
```

```
</TR>
```

```
</TABLE>
```

ДЕМОНСТРАЦИЯ COLSPAN И ЗАГОЛОВКОВ

He ad1	He ad2
A	B
C	D

```
<TABLE BORDER>
```

```
<TR>
```

```
<TH COLSPAN=2>Head1</TH>
```

```
<TH COLSPAN=2>Head2</TH>
```

```
</TR>
```

```
<TR>
```

```
<TD>A</TD> <TD>B</TD> <TD>C</TD> <TD>D</TD>
```

```
</TR>
```

```
<TR>
```

```
<TD>E</TD> <TD>F</TD> <TD>G</TD> <TD>H</TD>
```

```
</TR>
```

```
</TABLE>
```

ДЕМОНСТРАЦИЯ МНОЖЕСТВЕННЫХ ЗАГОЛОВКОВ, COLSPAN

Head1		Head2	
Hea d 3	Hea d 4	Hea d 5	Hea d 6
A	B	C	D
E	F	G	H

```
<TABLE BORDER>
```

```
<TR>
```

```
<TH COLSPAN=2>Head1</TH>
```

```

    <TH COLSPAN=2>Head2</TH>
  </TR>
  <TR>
    <TH>Head 3</TH> <TH>Head 4</TH>
    <TH>Head 5</TH> <TH>Head 6</TH>
  </TR>
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD> <TD>D</TD>
  </TR>
  <TR>
    <TD>E</TD> <TD>F</TD> <TD>G</TD> <TD>H</TD>
  </TR>
</TABLE>

```

ДЕМОНСТРАЦИЯ БОКОВЫХ ЗАГОЛОВКОВ

He ad1	Ite m 1	Ite m 2	Ite m 3
He ad2	Ite m 4	Ite m 5	Ite m 6
He ad3	Ite m 7	Ite m 8	Ite m 9

```

<TABLE BORDER>
  <TR><TH>Head1</TH>
    <TD>Item 1</TD> <TD>Item 2</TD> <TD>Item 3</TD></TR>
  <TR><TH>Head2</TH>
    <TD>Item 4</TD> <TD>Item 5</TD> <TD>Item 6</TD></TR>
  <TR><TH>Head3</TH>
    <TD>Item 7</TD> <TD>Item 8</TD> <TD>Item 9</TD></TR>
</TABLE>

```

ДЕМОНСТРАЦИЯ БОКОВЫХ ЗАГОЛОВКОВ, ROWSPAN

He ad1	Ite m 1	Item 2	Ite m 3	Item 4
	Ite m 5	Item 6	Ite m 7	Item 8
He ad2	Ite m 9	Item 10	Ite m 3	Item 11

```

<TABLE BORDER>
  <TR><TH ROWSPAN=2>Head1</TH>
    <TD>Item 1</TD> <TD>Item 2</TD> <TD>Item 3</TD> <TD>Item 4</TD>
  </TR>
  <TR><TD>Item 5</TD> <TD>Item 6</TD> <TD>Item 7</TD> <TD>Item 8</TD>
  </TR>
  <TR><TH>Head2</TH>
    <TD>Item 9</TD> <TD>Item 10</TD> <TD>Item 3</TD> <TD>Item 11
  </TD>
  </TR>
</TABLE>

```

ПРИМЕР ТАБЛИЦЫ ИСПОЛЬЗУЮЩЕЙ ВСЕ ЭТИ ТЭГИ

		Average	
		Height	Weight
Gender	Males	1.9	0.003
	Females	1.7	0.002

```

<TABLE BORDER>
  <TR>    <TD><TH ROWSPAN=2></TH>
          <TH COLSPAN=2>Average</TH></TD>
</TR>
<TR>    <TD><TH>Height</TH><TH>Weight</TH></TD>
</TR>
<TR>    <TH ROWSPAN=2>Gender</TH>
          <TH>Males</TH><TD>1.9</TD><TD>0.003</TD>
</TR>
<TR>    <TH>Females</TH><TD>1.7</TD><TD>0.002</TD>
</TR>
</TABLE>

```

ИСПОЛЬЗОВАНИЕ ROWSPAN/COLSPAN

A		
1		
2		
	3	
	4	
		I

```

<TABLE BORDER>
  <TR>
    <TD ALIGN=center ROWSPAN=2 COLSPAN=2>A</TD>
    <TD>1</TD>
    <TD>2</TD>
  </TR>
  <TR>
    <TD>3</TD>
    <TD>4</TD>
  </TR>
  <TR>
    <TD ALIGN=center ROWSPAN=2 COLSPAN=2>C</TD>
    <TD ALIGN=center ROWSPAN=2 COLSPAN=2>D</TD>
  </TR>
  <TR>
  </TR>
  </TR>
</TABLE>

```

**ВЫРАВНИВАНИЕ, ОТСТУПЫ, РАМКИ
ТАБЛИЦА БЕЗ РАМКИ**

Ite	Ite	Ite
m 1	m 2	m 3

Ite Ite
m 4 m 5

```
<TABLE border="0">
  <TR>   <TD>Item 1</TD> <TD ROWSPAN=2>Item 2</TD> <TD>Item 3
</TD>
  </TR>
  <TR>   <TD>Item 4</TD> <TD>Item 5</TD>
  </TR>
</TABLE>
```

ТАБЛИЦА С РАМКОЙ ШИРИНОЙ 10

Ite m 1	Ite m 2
Ite m 3	Ite m 4

```
<TABLE BORDER=10>
  <TR>   <TD>Item 1</TD> <TD> Item 2</TD>
  </TR>
  <TR>   <TD>Item 3</TD> <TD>Item 4</TD>
  </TR>
</TABLE>
```

CELLPADDING И CELLSPACING


```
<TABLE BORDER CELLPADDING=10 CELLSPACING=0>
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD>
  </TR>
  <TR>
    <TD>D</TD> <TD>E</TD> <TD>F</TD>
  </TR>
</TABLE>
```


```
<TABLE BORDER CELLPADDING=0 CELLSPACING=10>
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD>
  </TR>
  <TR>
    <TD>D</TD> <TD>E</TD> <TD>F</TD>
  </TR>
</TABLE>
```


```
<TABLE BORDER CELLPADDING=10 CELLSPACING=10>
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD>
  </TR>
  <TR>
    <TD>D</TD> <TD>E</TD> <TD>F</TD>
  </TR>
</TABLE>
```


```
<TABLE BORDER=5 CELLPADDING=10 CELLSPACING=10>
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD>
  </TR>
  <TR>
    <TD>D</TD> <TD>E</TD> <TD>F</TD>
  </TR>
</TABLE>
```

ВЫРАВНИВАНИЕ, ЗАГОЛОВКИ И ПОДТАБЛИЦЫ

ДЕМОНСТРАЦИЯ МНОЖЕСТВЕННЫХ СТРОК В ТАБЛИЦЫ СТРОК В ТАБЛИЦЕ

January	February	March
This is cell 1	Cell 2	Another cell, cell 3
Cell 4	and now this is cell 5	Cell 6

```
<TABLE BORDER>
  <TR>
    <TH>January</TH>
    <TH>February</TH>
    <TH>March</TH>
  </TR>
  <TR>
    <TD>This is cell 1</TD>
```

```

    <TD>Cell 2</TD>
    <TD>Another cell,<br> cell 3</TD>
</TR>
<TR>
    <TD>Cell 4</TD>
    <TD>and now this<br>is cell 5</TD>
    <TD>Cell 6</TD>
</TR>
</TABLE>

```

ALIGN=LEFT|RIGHT|CENTER

может применяться к отдельным ячейкам или всей строке

January	February	March
all aligned center	Cell 2	Another cell, cell 3
aligned right	aligned to center	default, aligned left

```

<TABLE BORDER>
  <TR>
    <TH>January</TH>
    <TH>February</TH>
    <TH>March</TH>
  </TR>
  <TR ALIGN=center>
    <TD>all aligned center</TD>
    <TD>Cell 2</TD>
    <TD>Another cell,<br> cell 3</TD>
  </TR>
  <TR>
    <TD ALIGN=right>aligned right</TD>
    <TD ALIGN=center>aligned to center</TD>
    <TD>default,<br>aligned left</TD>
  </TR>
</TABLE>

```

VALIGN=TOP|BOTTOM|MIDDLE

может применяться к отдельным ячейкам или всей строке

January	February	March
all aligned to top	and now this is cell 2	Cell 3
aligned to the top	aligned to the bottom	default alignment, center

```

<TABLE BORDER>
  <TR>
    <TH>January</TH>
    <TH>February</TH>

```

```

    <TH>March</TH>
</TR>
<TR VALIGN=top>
  <TD>all aligned to top</TD>
  <TD>and now this<br>is cell 2</TD>
  <TD>Cell 3</TD>
</TR>
<TR>
  <TD VALIGN=top>aligned to the top</TD>
  <TD VALIGN=bottom>aligned to the bottom</TD>
  <TD>default alignment,<br>center</TD>
</TR>
</TABLE>

```

CAPTION=TOP|BOTTOM

Верхний заголовок		
January	February	March
This cell 1	is Cell 2	Another cell, cell 3

```

<TABLE BORDER>
<CAPTION ALIGN=top>A top CAPTION</CAPTION>
<TR>
  <TH>January</TH>
  <TH>February</TH>
  <TH>March</TH>
</TR>
<TR>
  <TD>This is cell 1</TD>
  <TD>Cell 2</TD>
  <TD>Another cell,<br> cell 3</TD>
</TR>
</TABLE>

```

Нижний заголовок		
January	February	March
This cell 1	is Cell 2	Another cell, cell 3

```

<TABLE BORDER>
<CAPTION ALIGN=bottom>A bottom CAPTION</CAPTION>
<TR>
  <TH>January</TH>
  <TH>February</TH>
  <TH>March</TH>
</TR>

```

```

<TR>
  <TD>This is cell 1</TD>
  <TD>Cell 2</TD>
  <TD>Another cell,<br> cell 3</TD>
</TR>
</TABLE>

```

ВЛОЖЕННЫЕ ТАБЛИЦЫ: ТАБЛИЦА ABCD ВНУТРИ ТАБЛИЦЫ 123456

		3
		6

```

<TABLE BORDER>
  <TR> <!-- ROW 1, TABLE 1 -->
    <TD>1</TD>
    <TD>2</TD>
    <TD>3
      <TABLE BORDER>
        <TR> <!-- ROW 1, TABLE 2 -->
          <TD>A</TD>
          <TD>B</TD>
        </TR>
        <TR> <!-- ROW 2, TABLE 2 -->
          <TD>C</TD>
          <TD>D</TD>
        </TR>
      </TABLE>
    </TD>
  </TR>
  <TR> <!-- ROW 2, TABLE 1 -->
    <TD>4</TD>
    <TD>5</TD>
    <TD>6</TD>
  </TR>
</TABLE>

```

**ЗАДАНИЕ ШИРИНЫ ТАБЛИЦЫ
ШИРИНА 50%**

Width=50%	Width=50%
3	4

```

<TABLE BORDER WIDTH="50%">
  <TR><TD>Width=50%</TD><TD>Width=50%</TD>
  </TR>
  <TR><TD>3</TD><TD>4</TD>
  </TR>
</TABLE>

```

Item width affects cell size	
3	

```

<TABLE BORDER WIDTH="50%">
  <TR><TD>Item width affects cell size</TD><TD>2</TD>
</TR>
  <TR><TD>3</TD><TD>4</TD>
</TR>
</TABLE>

```

WIDTH=100%	This is item 2
3	4

```

<TABLE BORDER WIDTH="100%">
  <TR><TD>WIDTH=100%</TD><TD>This is item 2</TD>
</TR>
  <TR><TD>3</TD><TD>4</TD>
</TR>
</TABLE>

```

ЦЕНТРИРОВАНИЕ ТАБЛИЦЫ НА СТРАНИЦЕ

A	B	C
D	E	F

```

<CENTER>
<TABLE BORDER WIDTH="50%">
  <TR>
    <TD>A</TD> <TD>B</TD> <TD>C</TD>
  </TR>
  <TR>
    <TD>D</TD> <TD>E</TD> <TD>F</TD>
  </TR>
</TABLE>
</CENTER>

```

ШИРИНА ТАБЛИЦЫ И ВЛОЖЕННЫЕ ТАБЛИЦЫ

Item 1	Item 2	
Item A	Item B	Item 4

```

<TABLE BORDER WIDTH="50%">
  <TR><TD>Item 1</TD><TD>Item 2</TD>
</TR>
  <TR><TD>
    <TABLE BORDER WIDTH=100%>
      <TR><TD>Item A</TD><TD>Item B</TD>
    </TR>
    </TABLE>
  </TD>
  <TD>Item 4</TD>
</TR>
</TABLE>

```

HEIGHT=15%

HEIGHT=15%	Item 2
3	4

```

<TABLE BORDER WIDTH="50%" HEIGHT="15%">
  <TR><TD>HEIGHT=15%</TD> <TD>Item 2</TD>
</TR>
  <TR><TD>3</TD><TD>4</TD>
</TR>
</TABLE>

```

Задание 3. Фоновый рисунок

При помощи фильтров Photoshop создайте **5 (пять)** различных фоновых текстур для использования в качестве фона на Web-страницах. Размер текстур 50x50 пикселей. Старайтесь использовать в каждой текстуре различные фильтры. Используемые цвета должны соответствовать вашей палитре.

Создайте страницу **texture.htm** с образцами выполненных текстур. Эта страница может иметь такой вид. Разместите ее в разделе "**Учебная работа**" под пунктом меню "Текстуры". При формировании страницы используйте таблицы.

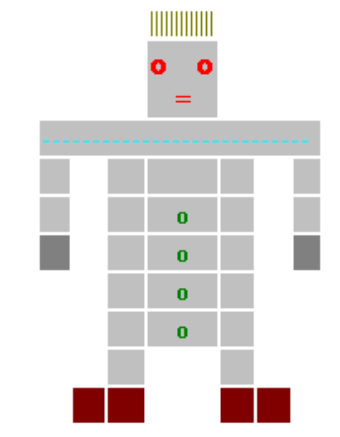
Задание 4. Заливка ячеек определенным цветом.

Для заливки ячейки таблицы используется атрибут **bicolor**. Откройте страницу, содержащую скриншоты вашей цветовой палитры сайта в режиме редактирования, и, опираясь на нее, создайте аналогичные таблицы, выполняя заливку ячеек соответствующими цветами. Набирать все теги вручную, так как при проверке задания будет учитываться чистота html-кода! Выполните на этой же странице задание.

Для четного варианта:

Для нечетного варианта:

Общее:



Указания по технике безопасности

При выполнении лабораторной работы **запрещается**:

1. Самостоятельно производить ремонт персонального компьютера, а так же, установку и удаление имеющегося программного обеспечения;
2. Нарушать общепринятые правила техники безопасности при работе с электрооборудованием, в частности, касаться электрических розеток металлическими предметами и т.д.;
3. Принимать пищу, напитки и сорить на рабочем месте пользователя персонального компьютера.

В случае неисправности персонального компьютера необходимо **немедленно** сообщить об этом обслуживающему персоналу лаборатории (системному администратору, оператору).

Содержание отчета и его форма

Отчет по лабораторной работе оформляется в виде текстового документа и должен включать:

1. Название практической работы.
2. Цель лабораторной работы.
3. Формулировку индивидуального задания и результат его выполнения.
4. Краткие выводы по результатам выполнения практической работы.

Защита практической работы

Оформленный, в соответствии с требованиями ГОСТов, отчет представляется студентом преподавателю для проверки и последующей защиты. Защита отчета по практической работе производится студентом только индивидуально.

В ходе защите практической работы студент отвечает на вопросы преподавателя (поясняет методику выполнения индивидуального задания, отвечает на контрольные вопросы и т.д.).

Отчет, оформленный с отступлениями от требований ГОСТов, небрежно и неаккуратно к защите не принимается.

Практическая работа № 18. Мультимедиа в html

Цель и содержание работы: Создать web-страницу «Мультимедиа», разместить на ней 3 звуковых файла, 3 видеофайла и 3 картинки.

Аппаратура и программное обеспечение.

Аппаратура. Для выполнения практической работы необходимы:

- персональный компьютер

- рабочие станции: монитор + клавиатура + мышь.

Программное обеспечение. Для выполнения лабораторной работы необходима операционная система WINDOWS 7/8/10 и подключение к глобальной сети Интернет.

Методика и порядок выполнения работы

Теоретический материал

Графика. Форматы интернет-графики

Графика на Web-страницах появилась достаточно давно. Предназначенный для этого тег появился еще в версии 3.2 языка HTML, которая вышла в 1997 году. С тех пор Всемирную паутину захлестнула волна интернет-графики (к настоящему времени, надо сказать, поутихшая).

Как уже говорилось, графические изображения — суть внедренные элементы Web-страниц. Это значит, что они сохраняются в отдельных от самой Web-страницы файлах.

Форматы интернет-графики

На данный момент существует несколько десятков форматов хранения графики в файлах. Но Web-обозреватели поддерживают далеко не все. В WWW сейчас используются всего три формата, которые мы далее рассмотрим.

Нужно отметить, что все три формата поддерживают сжатие графической информации. Благодаря сжатию размер графического файла сильно уменьшается, и поэтому он передается по сети быстрее, чем несжатый файл.

Формат **GIF (Graphics Interchange Format, формат обмена графикой)** — старожил среди "сетевых" форматов графики. Он был разработан еще в 1987 году и модернизирован в 1989 году. На данный момент он считается устаревшим, но все еще широко распространен.

Достоинств у него довольно много. Во-первых, GIF позволяет задать для изображения "прозрачный" цвет; закрашенные этим цветом области изображения станут своего рода "дырками", сквозь которые будет "просвечивать" фон родительского элемента. Во-вторых, в одном GIF-файле можно хранить сразу несколько изображений, фактически — настоящий фильм (анимированный GIF). В-третьих, из-за особенностей применяемого в нем сжатия он отлично подходит для хранения штриховых изображений (карт, схем, рисунков карандашом и пр.).

Недостаток у формата GIF всего один — он совершенно не годится для хранения полутоновых изображений (фотографий, картин и т. п.). Это обусловлено тем, что GIF-изображения могут включать всего 256 цветов, и потерями качества при сжатии.

GIF используется для хранения элементов оформления Web-страниц (всяческих линеек, маркеров списков и т. п.) и штриховых изображений.

Формат **JPEG (Joint Photographic Experts Group, Объединенная группа экспертов по фотографии)** был разработан в 1993 году специально для хранения полутоновых изображений. Для чего активно применяется до сих пор.

JPEG, в отличие от GIF, не ограничивает количество цветов у изображения, а реализованное в нем сжатие лучше всего подходит для полутоновых изображений. Однако он плохо справляется с штриховой графикой, не поддерживает "прозрачный" цвет и анимацию.

Формат **PNG (Portable Network Graphics, перемещаемая сетевая графика)** появился на свет в 1996 году. Он разрабатывался как замена устаревшему и не очень удобному GIF, а также, в некоторой степени, JPEG. В настоящее время он последовательно отвоевывает "жизненное пространство" у GIF.

К достоинствам формата PNG можно отнести возможность хранения как штриховых, так и полутоновых изображений и поддержку полупрозрачности. Недостаток всего один и некритичный — невозможность хранения анимации.

Осталось назвать расширения, под которыми сохраняются файлы того или иного формата. Файлы GIF и PNG имеют "говорящие" расширения gif и png, а файлы JPEG — jpg, jpe или jpeg.

Вставка графических изображений

Добавить на Web-страницу графическое изображение позволяет одинарный тег ****. Web-обозреватель поместит изображение в том месте Web-страницы, в котором встретился тег ****.

Обязательный атрибут тега **SRC** служит для указания интернет-адреса файла с изображением.

Пример:

```
<IMG SRC="image.gif">
```

Этот тег помещает на Web-страницу изображение, хранящееся в файле image.gif, который находится в той же папке, что и файл самой этой Web-страницы.

Пример:

```
<IMG SRC="/images/picture.jpg">
```

Данный тег помещает на Web-страницу изображение, хранящееся в файле picture.jpg, который находится в папке images, вложенной в корневую папку Web-сайта.

Пример:

```
<IMG SRC="http://www.othersite.ru/book12/author.jpg">
```

А этот тег помещает на Web-страницу изображение, хранящееся в файле с интернет-адресом <http://www.othersite.ru/book12/author.jpg>, т. е. изображение с другого Web-сайта.

Рассмотрим еще один атрибут тега ****, который может нам пригодиться в дальнейшем.

Поскольку изображение хранится в отдельном от Web-страницы файле, Web-обозревателю придется послать Web-серверу еще один запрос на его получение. Web-серверу нужно найти этот файл и отправить его Web-обозревателю. Файл должен загрузиться по сети.

На все это требуется время. Если изображений на Web-странице много, все они велики по размеру, а канал связи медленный, понадобится значительное время. Может случиться так, что сама Web-страница будет успешно загружена и отображена на экране, а изображения — еще нет. И Web-обозреватель вместо не загруженного еще изображения выведет на экран пустой прямоугольник.

Возникают две проблемы. Во-первых, пустые прямоугольники вместо изображений выглядят некрасиво. Во-вторых, посетитель не сможет понять, что за изображение должно находиться вместо того или иного прямоугольника, и стоит ли ждать окончания его загрузки.






И если с первой проблемой справиться практически невозможно, то вторую мы вполне способны решить. Для этого тег поддерживает необязательный атрибут **ALT**, с помощью которого указывается так называемый текст замены. Он будет отображаться в пустом прямоугольнике, обозначающем незагруженное изображение, пока это изображение не загрузится:

```
<P><IMG SRC="image.gif" ALT="Пример изображения"></P>
```

Вот еще несколько необязательных атрибутов:

- **height**-Высота изображения
- **width**-Ширина изображения
- **align**-Определяет как рисунок будет выравниваться по краю

и способ обтекания текстом

Табл. 1. Использование значений атрибута align		
Значение	Описание	Пример
bottom	Выравнивание нижней границы изображения по окружающему тексту..	 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
left	Выравнивает изображение по левому краю окна.	 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
middle	Выравнивание середины изображения по базовой линии текущей строки.	 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
right	Выравнивает изображение по правому краю окна.	 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
top	Верхняя граница изображения выравнивается по самому высокому элементу текущей строки.	 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>

- **border**-Толщина рамки вокруг изображения

Мультимедиа. Форматы файлов и форматы кодирования

Мультимедиа — это, в первую очередь, аудио и видео. Мультимедиа в приложении к Web-дизайну — это аудио- и видеоролики, размещенные на Web-страницах.

До недавних пор разместить на Web-странице аудио- или видеоролик можно было только с помощью громоздкого HTML-кода, дополнительных программ и "шаман-ских плясок" вокруг всего этого. Но сейчас, с появлением HTML 5 и поддерживающих его (хотя бы частично) Web-обозревателей, потребуется всего один тег. Какой? Очень простой, не сложнее уже знакомого нам тега !

Форматы файлов и форматы кодирования

Форматов мультимедийных файлов существует не меньше, чем форматов файлов графических. Как и в случае с интернет-графикой, Web-обозреватели поддерживают далеко не все мультимедийные форматы, а только немногие. (Хотелось бы автору посмотреть на Web-обозреватель, который поддерживает все форматы файлов — и на сам Web-обозреватель, и на его размеры...)

Но Web-обозревателю мало поддерживать только сам формат мультимедийных файлов. Он должен быть "знаком" и с форматом кодирования записанной в нем аудио- и (или) видеоинформации. В мире мультимедиа так — разные файлы одного формата могут хранить информацию, закодированную разными форматами. Более того, аудио- и видеодорожки мультимедийного файла практически всегда кодируются разными форматами.

Практически все форматы кодирования мультимедийных данных поддерживают их сжатие. Благодаря этому размер мультимедийных файлов значительно (иногда на несколько порядков) уменьшается, что благотворно сказывается на скорости их передачи по сети.

Перечислим и кратко опишем все форматы мультимедийных файлов, используемые в Web-дизайне и поддерживаемые Web-обозревателями.

- **WAV (WAVE, волна)** — "старожил" среди мультимедийных форматов. Был разработан Microsoft в самом начале 90-х годов прошлого века для хранения аудио-данных и применяется для этой цели до сих пор. Файлы такого формата имеют расширение wav.

- **OGG** — более новый формат. Был разработан около десяти лет назад некоммерческой организацией Xiph.org для хранения аудио- и видеоинформации. Файлы этого формата имеют расширения ogg (универсальное расширение), oga (аудио-файлы) и ogv (видеофайлы); последние два расширения встречаются редко.

- **MP4** — также "новичок". Был разработан организацией Motion Picture Expert Group (Экспертная группа по вопросам движущегося изображения; также известна как MPEG) в 1998 году для хранения аудио- и видеоданных. Файлы этого формата имеют расширение mp4.

- **QuickTime** — формат очень старый, он старше даже WAV. Был разработан Apple в 1989 году для хранения аудио- и видеоданных. Файлы такого формата имеют расширение mov.

Теперь рассмотрим форматы кодирования аудио и видео, поддерживаемые современными Web-обозревателями.

- **PCM (Pulse-Coded Modulation, импульсно-кодовая модуляция)** — самый простой и самый старый формат кодирования. Он даже не поддерживает сжатие информации. Служит для кодирования аудиоданных.

- **Vorbis** — более современный формат кодирования. Был представлен организацией Xiph.org (разработчиком формата файла OGG) в 2002 году. Используется для кодирования аудиоданных.

- **AAC (Advanced Audio Coding, развитие кодирование аудио)** — не очень новый формат кодирования. Был разработан организацией Motion Picture Expert Group в 1997 году. Применяется для кодирования аудиоданных.

- **Theora** — пожалуй, самый "молодой" формат кодирования. Он также был разработан организацией Xiph.org несколько лет назад. Используется для кодирования видеоданных.

- **H.264** — тоже очень "молод". Был представлен организациями Motion Picture Expert Group и Video Coding Experts Group (Группа экспертов по кодированию видео) в 2003 году. Предназначен для кодирования видеоданных.

Почти все эти форматы являются открытыми. Исключения — формат файлов QuickTime, принадлежащий Apple, и формат кодирования H.264, защищенный более чем сотней патентов.

Типы MIME

По сети передаются самые разные данные: Web-страницы, графические изображения, аудио- и видеофайлы, архивы, исполняемые файлы и пр. Эти данные предназначены разным программам. К тому же, с разными данными программа, принявшая их, может поступить по-разному. Так, Web-обозреватель при получении Web-страницы или графического изображения отобразит их на экране, а при получении архива или исполняемого файла — откроет или сохранит его на диске.

Всем передаваемым по сети данным присваивается особое обозначение, однозначно указывающее на их природу, — тип MIME (Multipurpose Internet Mail Extensions, многоцелевые расширения почты Интернета). Тип MIME присваивает данным программа, их отправляющая, например, Web-сервер. А принимающая программа (тот же Web-обозреватель) по типу MIME принятых данных определяет, поддерживает ли она эти данные, и, если поддерживает, что с ними делать.

Выполнить задание

Вставка аудиоролика

Для вставки на Web-страницу аудиоролика язык HTML 5 предусматривает парный тег `<AUDIO>`. Интернет-адрес файла, в котором хранится этот аудиоролик, указывают с помощью атрибута **SRC** этого тега: `<AUDIO SRC="sound.wav"></AUDIO>`

Встретив тег `<AUDIO>`, Web-обозреватель может сразу загрузить и воспроизвести аудиофайл, только загрузить его без воспроизведения или вообще ничего не делать. Также он может вывести на Web-страницу элементы управления, с помощью которых посетитель запускает воспроизведение аудиофайла, приостанавливает его, прокручивает вперед или назад и регулирует громкость. Все это настраивается с помощью различных атрибутов тега `<AUDIO>`.

Тег `<AUDIO>` создает блочный элемент Web-страницы. Так что мы не сможем вставить аудиоролик на Web-страницу в качестве части абзаца. Зато, чтобы поместить его в отдельный абзац, нам не придется совершать никаких дополнительных действий (в отличие от изображения).

По умолчанию Web-обозреватель не будет воспроизводить аудиоролик. Чтобы он это сделал, в теге `<AUDIO>` нужно указать особый атрибут **AUTOPLAY**. Это действительно особый атрибут: он не имеет значения — достаточно одного его присутствия в теге, чтобы он начал действовать (атрибут тега без значения):

<P>Сейчас вы услышите звук!</P>
<AUDIO SRC="sound.ogg" AUTOPLAY></AUDIO>

По умолчанию аудиоролик никак не отображается на Web-странице (что, впрочем, понятно — аудио нужно не смотреть, а слушать). Но если в теге <AUDIO> поставить атрибут без значения **CONTROLS**, Web-обозреватель выведет в том месте Web-страницы, где проставлен тег <AUDIO>, элементы управления воспроизведением аудиоролика. Они включают кнопку запуска и приостановки воспроизведения, шкалу воспроизведения и регулятор громкости:

<P>Нажмите кнопку воспроизведения, чтобы услышать звук.</P>
<AUDIO SRC="sound.ogg" CONTROLS></AUDIO>

Атрибут без значения **AUTOBUFFER** имеет смысл указывать в теге <AUDIO> только в том случае, если там отсутствует атрибут AUTOPLAY. Если он указан, Web-обозреватель сразу после загрузки Web-страницы начнет загружать файл аудиоролика — это позволит исключить задержку файла перед началом его воспроизведения. Ниже представлен пример вставки аудиоролика

```
<!DOCTYPE html>
<HTML>
  <HEAD>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1251" >
    <TITLE>Тег AUDIO</TITLE>
  </HEAD>
  <BODY>
    <H1>Тег AUDIO</H1>
    <P>Тег AUDIO служит для вставки на Web-страницу аудиоролика.</P>
    <H6>Пример:</H6>
    <PRE><AUDIO SRC="sound.wav"
    CONTROLS></AUDIO></PRE>
    <H6>Результат:</H6>
    <AUDIO SRC="sound.wav" CONTROLS></AUDIO>
  </BODY>
</HTML>
```

В результате получите:



Вставка видеоролика

Для вставки на Web-страницу видеоролика предназначен парный тег **<VIDEO>**. Интернет-адрес видеофайла указывается с помощью знакомого нам атрибута **SRC** этого тега:

```
<VIDEO SRC="film.ogg"></VIDEO>
```

Встретив этот тег, Web-обозреватель выведет в том месте Web-страницы, где он проставлен, панель просмотра содержимого видеоролика. В зависимости от указанных нами в теге **<VIDEO>** атрибутов, он может сразу загрузить и воспроизвести аудиофайл, только загрузить его без воспроизведения или вообще ничего не делать. Также он может вывести на Web-страницу элементы управления воспроизведением видеоролика.

Как и тег **<AUDIO>**, тег **<VIDEO>** создает блочный элемент Web-страницы и поддерживает атрибуты **AUTOPLAY**, **CONTROLS** и **AUTOBUFFER**:

```
<VIDEO SRC="film.ogg" AUTOPLAY CONTROLS></VIDEO>
```

Если воспроизведение видеоролика еще не запущено, в панели просмотра будет выведен первый его кадр или вообще ничего (конкретное поведение зависит от Web-обозревателя). Но мы можем указать графическое изображение, которое будет туда выведено в качестве заставки. Для этого служит атрибут **POSTER** тега **<VIDEO>**; его значение указывает интернет-адрес нужного графического файла:

```
<VIDEO SRC="film.ogg" CONTROLS POSTER="filmposter.jpg"></VIDEO>
```

Здесь мы указали для видеоролика изображение-заставку, которое будет выведено в панели просмотра перед началом его воспроизведения и которое хранится в файле `filmposter.jpg`. Ниже представлен пример вставки видеоролика

```
<!DOCTYPE html>
<HTML>
  <HEAD>
    <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1251" />
    <TITLE>Ter VIDEO</TITLE>
  </HEAD>
  <BODY>
    <H1>Ter VIDEO</H1>
    <P>Ter VIDEO служит для вставки на Web-страницу видеоролика.</P>
    <H6>Пример:</H6>
    <PRE><VIDEO SRC="film.ogg"
    CONTROLS></VIDEO></PRE>
    <H6>Результат:</H6>
    <VIDEO SRC="film.ogg" CONTROLS></VIDEO>
  </BODY>
</HTML>
```



В результате получите:

Указания по технике безопасности

При выполнении лабораторной работы **запрещается**:

4. Самостоятельно производить ремонт персонального компьютера, а так же, установку и удаление имеющегося программного обеспечения;
5. Нарушать общепринятые правила техники безопасности при работе с электрооборудованием, в частности, касаться электрических розеток металлическими предметами и т.д.;
6. Принимать пищу, напитки и сорить на рабочем месте пользователя персонального компьютера.

В случае неисправности персонального компьютера необходимо **немедленно** сообщить об этом обслуживающему персоналу лаборатории (системному администратору, оператору).

Содержание отчета и его форма

Отчет по лабораторной работе оформляется в виде текстового документа и должен включать:

5. Название практической работы.
6. Цель практической работы.
7. Формулировку индивидуального задания и результат его выполнения.
8. Краткие выводы по результатам выполнения практической работы.

Защита практической работы

Оформленный, в соответствии с требованиями ГОСТов, отчет представляется студентом преподавателю для проверки и последующей защиты. Защита отчета по практической работе производится студентом только индивидуально.

В ходе защите практической работы студент отвечает на вопросы преподавателя (поясняет методику выполнения индивидуального задания, отвечает на контрольные вопросы и т.д.).

Отчет, оформленный с отступлениями от требований ГОСТов, небрежно и неаккуратно к защите не принимается.