

Лабораторная работа № 1

Дисциплина “Программные средства создания Internet-приложений”

Специальности “Программное обеспечение информационных технологий”

Тема: Создание и форматирование простейшего HTML-документа

Цель работы: Научиться создавать простейший html-документ

Средства: ПЭВМ, Notepad.

1 Вопросы для самоконтроля

1. Что такое браузер?
2. Что такое HTML?
3. Кто разработал HTML? Где? В каком году?
4. Приведите примеры объектов документа.
5. При помощи каких конструкций описываются объекты документа на языке HTML?
6. Приведите общий вид записи открывающего тега.
7. Приведите общий вид записи закрывающего тега.
8. Какое назначение имеет имя тега?
9. Для чего служат атрибуты тега?
10. Назовите две формы задания элемента на языке HTML при помощи тегов.
11. В каком регистре нужно записывать имена тегов, имена атрибутов и их значения?
12. Какой символ должен отделять имя тега от списка атрибутов?
13. Каким знаком разделяются между собой атрибуты?
14. Какой символ отделяет имя атрибута от его значения?
15. Можно ли вместо одного пробела-разделителя использовать несколько?
16. Можно ли переносить запись тега в позиции разделителя-пробела на другую строку?

2 Ход работы:

1. Изучите теоретические сведения
2. Выполните практические задания
3. Напишите отчет

3 Теоретические сведения

Краткие теоретические сведения

Создание простейшего web-документа

Основное, что нужно знать для того, чтоб сделать простую html-страничку, это HTML (от Hypertext Markup Language — язык разметки гипертекста). А пишется этот язык специальными знаками (тегами), которые находятся в угловых скобках, как вот этот тег **<html>** он означает, что началась страничка с кодировкой, он обязателен в любой интернет страничке. У большинства тегов есть заканчивающий тег. Он выглядит так **</тег>** он нужен для определения того места, где действие тега закончилось.

<тег>действие тега</тег>

Любая страничка (документ) в Интернете, написанная на гипертекстовой разметке, содержит голову и тело. Голова обозначается тегом **<head>**, а тело **<body>**.

Вот простейший пример интернет странички в кодировке **html** с обязательными тегами, которые присутствуют в любой страничке:

Пример 1. Пример странички в кодировке **html** с обязательными тегами

<html>	<!--Начало документа -->
<head>	<!--Начало заголовка -->
<title>	<!--Начало названия документа -->
Моя первая Web-страница	
</title>	<!--Конец названия документа -->
</head>	<!-- Конец заголовка -->
<body>	<!--Начало тела документа -->
Привет!	
</body>	<!--Конец тела документа -->
</html>	<!--Конец документа -->

Всё, что находится в угловых скобках с восклицательным знаком
<!-- комментарий-->

Многие теги, помимо имени, могут содержать **атрибуты** — элементы, дающие дополнительную информацию о том, как браузер должен обработать текущий тег. В нашем простейшем документе, однако, нет ни одного атрибута.

Чтобы созданный вами сайт могли легко найти поисковые системы (по ключевому слову, описанию, имени автора и т. д.), в разделе `<head>...</head>` размещается тег `<meta>`, в качестве значения атрибута **name** которого указывается имя некоторого свойства, а в качестве значения атрибута **content** — значение этого свойства:

- **keywords** (ключевые слова) — слова, которые, скорее всего, будут использованы для поиска вашей web-страницы;
- **description** (описание) — описание вашей web-страницы (ее краткая аннотация);
- **author** (имя автора web-страницы) — ваше имя.

Пример 2. Использование тега `<meta>`

```
<html>
<head>
<title>Практическая работа 2</title>
<meta name= "keywords" content = "Программист">
<meta name= "description" content = "Первые уроки создания
сайта">
<meta name= "author" content = "Иванов Пётр">
</head>
<body>
</body>
</html>
```

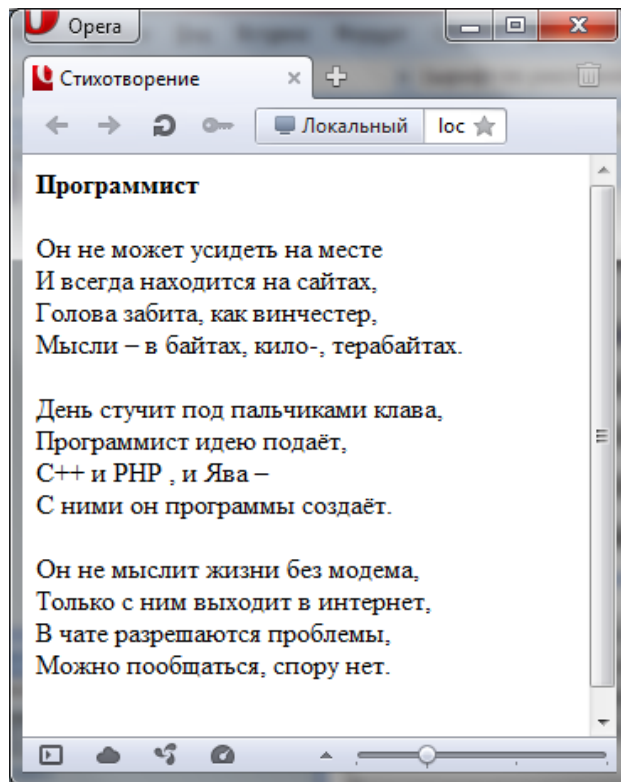
Принудительный разрыв строки

Тег `
` заставляет браузер продолжить вывод абзаца с новой строки. Тег `
` не имеет парного.

Посмотрите, как теперь выглядят стихи.

Пример 3. Принудительный разрыв строки

```
<html>
<head>
<title>Стихотворение</title>
</head>
<body>
<strong>Программист</strong>
<br><br>
Он не может усидеть на месте<br>
И всегда находится на сайтах,<br>
Голова забита, как винчестер,<br>
Мысли – в байтах, кило-, терабайтах.<br><br>
День стучит под пальчиками клавиша,<br>
Программист идею подаёт,<br>
С++ и PHP , и Ява –<br>
С ними он программы создаёт. <br><br>
Он не мыслит жизни без модема,<br>
Только с ним выходит в интернет,<br>
В чате разрешаются проблемы,<br>
Можно пообщаться, спору нет.<br><br><br>
</body>
</html>
```



Заголовки

Заголовки являются важной частью документа. Они разделяют информацию на отдельные логические части и поэтому существенно улучшают её восприятие.

HTML поддерживает заголовки шести уровней при помощи конструкции:

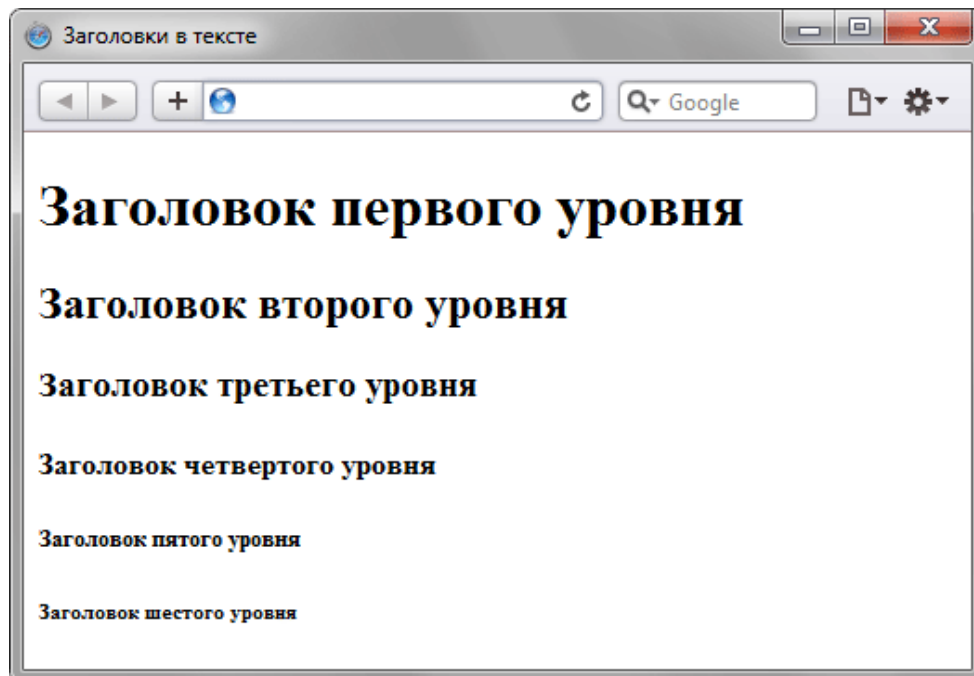
`<hn>текст заголовка</hn>`

В качестве **n** следует использовать одну из цифр 1–6, обозначающую уровень заголовка. При отображении страницы браузер выделяет заголовки размером шрифта и жирностью:

Элементом **h1** задают, как правило, заголовок сайта, элементом **h2** — заголовок страницы, элементом **h3** — разделы на странице и так далее.

Пример 4. Заголовки

```
<html>
<head>
  <title>Заголовки в тексте</title>
</head>
<body>
  <h1>Заголовок первого уровня</h1>
  <h2>Заголовок второго уровня</h2>
  <h3>Заголовок третьего уровня</h3>
  <h4>Заголовок четвертого уровня</h4>
  <h5>Заголовок пятого уровня</h5>
  <h6>Заголовок шестого уровня</h6>
</body>
</html>
```



Горизонтальная линия

Линия — простейший графический элемент, который умеет строить браузер. Горизонтальная линия задаётся тегом **<hr>**, и этот тег не имеет парного закрывающего.

Имеет ряд атрибутов, **align** -выравнивание с одним из трёх значений (**center**, **left**, **right**) может быть применен, если задана длина линии атрибут **width** в пикселях или процентах. Так же можно задать толщину линии атрибут - **size**, цвет атрибут - **color**, и при необходимости отключить тень линии **noshade**.

Пример 5. Горизонтальные линии

```
<html>
<head>
<title>Горизонтальная линия</title>
</head>
<body>
Это просто линия по умолчанию: <hr>;
<hr>
Это линия без тени: <hr noshade>;
<hr noshade>
Это линия окрашенная в красный цвет: <hr color="#ff0000">;
<hr color="#ff0000">
Линия длиной 250 пикселей: <hr width="250">;
<hr width="250">
Линия длиной 250 и толщиной 5 пикселей: <hr width="250" size="5">;
<hr width="250" size="5">
Линия длиной 500, толщиной 50, синяя: <hr width="500" size="50"
color="#0000ff">;
<hr width="500" size="50" color="#0000ff">
Примеры выравнивания:
<hr align="left" width="250" size="5">
<hr align="center" width="250" size="5">
<hr align="right" width="250" size="5">
<hr>
</body>
</html>
```

Это просто линия по умолчанию: <hr>	
Это линия без тени: <hr noshade >	
Это линия окрашенная в красный цвет: <hr color="#ff0000">	
Линия длиной 250 пикселей: <hr width="250">	
Линия длиной 250 и толщиной 5 пикселей: <hr width="250" size="5">	
Линия длиной 500, толщиной 50, синяя: <hr width="500" size="50" color="#0000ff">	
Примеры выравнивания:	

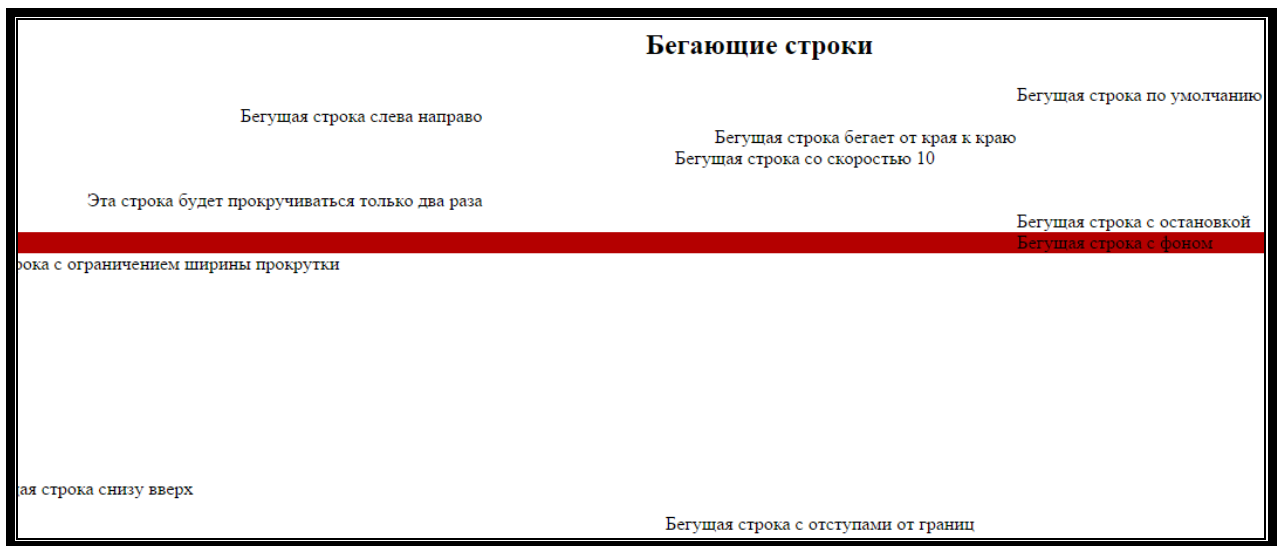
Бегущая строка

Тег **<marquee>** заставляет текст помещённый в него двигаться в том или ином направлении, проще говоря делает его бегущей строкой. Бегущая строка имеет ряд настроек скроллинга, которые задаются следующими атрибутами:

- behavior** - определяет тип скроллинга, может иметь следующие значения:
- alternate** - колебательные движения от края к краю
- scroll** - прокручивание текста. текст будет выходить за рамки экрана и снова появляться с противоположной его стороны
- slide** - прокручивание текста с остановкой.
- scrollamount** - скорость бегущей строки от 1 до 10.
- loop** задает количество прокруток бегущей строки.
- direction** - направление движения текста. значения:
 - up** - вверх,
 - down** - вниз,
 - left** - влево,
 - right** - вправо.
- bgcolor** - цвет фона бегущей строки,
- height** - высота строки,
- width** - ширина строки.

Пример 6. Бегущая строка

```
<html>
<head>
<title>Бегущая строка</title>
</head>
<body>
<div align="center"><h2>Бегающие строки</h2></div>
<marquee>Бегущая строка по умолчанию</marquee>
<marquee direction="right">Бегущая строка слева направо</marquee>
<marquee behavior="alternate">Бегущая строка бежит от края к краю</marquee>
<marquee scrollamount="10">Бегущая строка со скоростью 10</marquee>
<marquee scrollamount="1">Бегущая строка со скоростью 1</marquee>
<marquee direction="right" loop="2">Эта строка будет прокручиваться только два
раза</marquee>
<marquee behavior="slide">Бегущая строка с остановкой</marquee>
<marquee bgcolor="#b40000">Бегущая строка с фоном</marquee>
<marquee width=400>Бегущая строка с ограничением ширины прокрутки</marquee>
<marquee direction="up">Бегущая строка снизу вверх</marquee>
<marquee hspace="300">Бегущая строка с отступами от границ</marquee>
</body>
</html>
```



Абзацы и расположение абзацев

`<p>...</p>` - такая пара тегов описывает абзац. Все, что заключено между `<p>` и `</p>`, воспринимается как один абзац.

Теги `<h>` и `<p>` могут содержать дополнительный атрибут **align**, обычно указывается одно из трех значений этого атрибута:

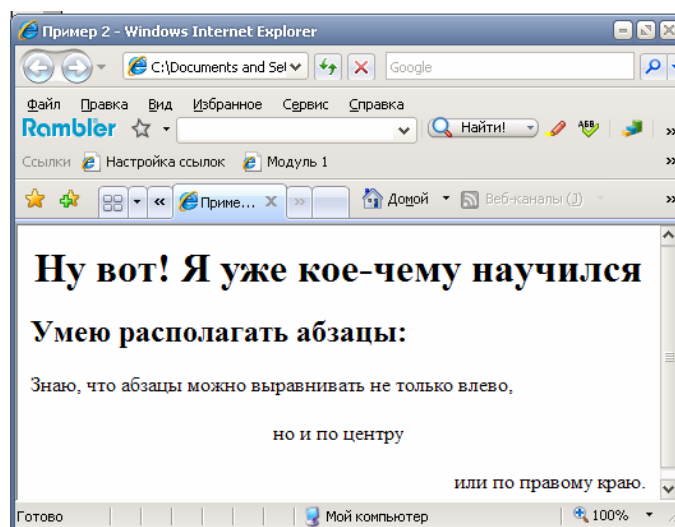
align=center – выравнивание по центру;

align=left – выравнивание по левому краю;

align=right – выравнивание по правому краю.

Пример 7. Расположение абзацев

```
<html>
<head>
<title>Пример</title>
</head>
<body>
<h1 align=center>Ну вот! Я уже кое-чему научился</h1>
<h2>Умею располагать абзацы: </h2>
<p>Знаю, что абзацы можно выравнивать не только влево, </p>
<p align=center>но и по центру</p> <p align=right>или по правому краю.</p>
</body>
</html>
```



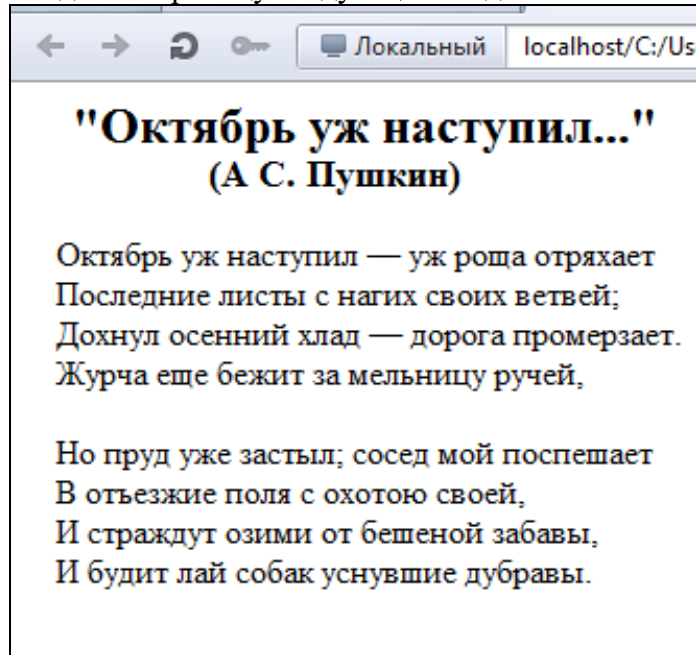
Задания для самостоятельного выполнения:

Задание 1.

Выполните примеры 2.3-2.7.

Задание 2.

Создайте страницу следующего вида



Задание 3.

Приведенный ниже текст оформите в виде HTML-документа.

«Язык разметки гипертекста HTML»

1. Каким тегом объявляется web-страница?

<html> </html>

2. Каким тегом объявляется заголовок web-страницы?

<head> </head>

3. В какой тег заключается основное содержание web-страницы?

<body> </body>

4. В какой тег заключается название web-страницы?

<title> </title>

5. Какой код для пустой web-страницы правильный?

<html> <head> <title> </title> </head> <body> </body> </html>

Задание 4.

Составить свое резюме в виде HTML-страницы по следующему плану (в общей сложности 4 – 5 строчек, вместо многоточий и пояснений в скобках вставить свои данные). Используйте в резюме горизонтальные линии, бегущие строки, различные заголовки и выравнивания текста.

Я, ... (ФИО), родился ... (дата и год рождения) в городе ... (место, где родились).

В 20... году окончил ... школу (лицей, гимназию) с углубленным изучением ..., средний балл аттестата ...

В 20... году поступил в МТЭК Белкоопсоюза на специальность Средний балл по первым сессиям (аттестациям) ...

В свободное от учебы время увлекаюсь ... (перечислить не менее трех увлечений помимо учебы).

Мои приятели в группе: ...(перечислить не менее трех фамилий из группы).

Задание 5(по вариантам).

Используя сведения из биографии известных людей в области программирования, создайте простейшие html-страницы. На странице необходимо обязательно использовать различные заголовки, расположение абзацев, линии, бегущие строки.

1. **Дональд Эрвин Кнут** (англ. Donald Ervin Knuth, родился 10 января 1938) — американский учёный, почётный профессор Стэнфордского университета и нескольких

других университетов в разных странах, иностранный член Российской академии наук, преподаватель и идеолог программирования, автор 19 монографий (в том числе ряда классических книг по программированию) и более 160 статей, разработчик нескольких известных программных технологий.

Автор всемирно известной серии книг, посвящённой основным алгоритмам и методам вычислительной математики, а также создатель настольных издательских систем TEX и METAFONT, предназначенных для набора и вёрстки книг, посвящённых технической тематике (в первую очередь — физико-математических).

2. **Сэр Тимоти Джон Бернерс-Ли** — знаменитый британский учёный, изобретатель URI, URL, HTTP, HTML, изобретатель Всемирной паутины (совместно с Робертом Кайо) и действующий глава Консорциума Всемирной паутины. Автор концепции семантической паутины. Автор множества других разработок в области информационных технологий.

Тим Бернерс-Ли родился 8 июня **1955** года в Лондоне (Англии). Его родители, Конвэй Бернерс-Ли и Мэри Ли Вудс оба были математиками и трудились над созданием «Manchester Mark I», одного из первых компьютеров. Тим учился в школе Эмануэль в городе Вэндсворте, затем в Королевском колледже в Оксфорде. Там он собрал свой первый компьютер на базе процессора M6800 с телевизором вместо монитора.

После окончания Оксфордского университета в **1976** году Бернерс-Ли поступил на работу в компанию «Plessey Telecommunications Ltd» в графстве Дорсет. Там он проработал два года, занимаясь в основном системами распределённых транзакций.

3. Родился **Марк Цукерберг** 14 мая **1984** года, в Соединенных Штатах Америки, в обычной еврейско-американской семье врачей (отец - стоматолог, мать - психиатр). Он не был единственным ребенком в семье, помимо нашего героя у Цукербергов еще трое отпрысков.

Детство Марка протекало, в общем-то, довольно обычно, до тех пор, пока ему не купили компьютер. Дело в том, что у одного его друга компьютер уже имелся, и, находясь под впечатлением, Марк уговорил родителей купить технику и ему. Произошло радостное событие, когда Цукербергу было 10-12 лет (он сам затрудняется вспомнить точнее), и машинка представляла собой Quantex 486DX на процессоре Intel 486. После приобретения Марк почувствовал себя ужасно взрослым и от новой игрушки первое время буквально не отходил. Впрочем, уже через пару месяцев ему надоело просто менять цвет бэкграунда, и он принялся читать умные книжки, решив научиться чему-то более полезному, а именно – [программированию](#).

4. **Алан Кэй** (Alan Curtis Kay; род. 17 мая 1940) — американский учёный в области теории вычислительных систем. Один из пионеров в областях объектно-ориентированного программирования и графического интерфейса.

Разработал язык программирования Smalltalk, где впервые был применен объектно-ориентированный подход.

Лауреат премии Тьюринга 2003 года за работу над объектно-ориентированным программированием, Премии Киото (2004).

"Я придумал термин «объектно-ориентированный», и вот что я вам скажу, я не имел ввиду C++." Кей Алан.

Один из создателей проекта One Laptop Per Child.

5. **Юкихиро Мацумото** (также известный как Matz, род. 14 апреля 1965) — японский разработчик свободного ПО, создатель языка программирования Ruby.

В интервью «Japan Inc.» он говорил, что сам учился программировать ещё до окончания школы. Он закончил университет города Цукуба, где он занимался исследованиями языков программирования и компиляторов.

"Я хочу, чтобы компьютер был моим слугой, а не господином, поэтому я должен уметь быстро и эффективно объяснить ему, что делать." Мацумото Юкихиро.

С 2006 года возглавляет отдел исследований и разработок Network Applied

Communication Laboratory, японский системный интегратор свободного ПО.

Родился в 1965 в префектуре Осака, но в возрасте четырёх лет переехал в город Ёнаго префектуры Тоттори, поэтому часто представляется как уроженец Ёнаго. В настоящее время проживает в городе Мацуэ префектуры Симанэ.

6. Эндрю Стюарт Таненбаум (англ. Andrew Stuart Tanenbaum) (родился в 1944 году) — профессор Амстердамского свободного университета, где возглавляет группу разработчиков компьютерных систем, доктор наук.

Известен как автор Minix (свободная Unix-подобная операционная система для студенческих лабораторий), книг по компьютерным наукам и RFID-вируса. Также является главным разработчиком пакета «Amsterdam Compiler Kit». Сам он считает свою преподавательскую деятельность наиболее важной.

"Не думай свысока о скорости передачи данных автомобилем, полным кассет, с грохотом передвигающимся по дороге." Таненбаум Эндрю.

Эндрю Таненбаум родился в Нью-Йорке и вырос в Уайт Плэйнс, штата Нью-Йорк. Получил учёную степень бакалавра по физике в MIT в 1965 году, также получил степень доктора физики в Калифорнийском университете Беркли в 1971 году.

7. Мартин Фаулер (англ. Martin Fowler) — автор ряда книг и статей об архитектуре ПО, объектно-ориентированному анализу и разработке, языку UML, рефакторингу, экстремальному программированию.

Родился в Англии, жил в Лондоне до переезда в Америку в 1994 г. В настоящее время живёт в Бостоне, штат Массачусетс.

"Любой дурак может написать программу, которую поймет компилятор. Хорошие программисты пишут программы, которые смогут понять другие программисты." Фаулер Мартин.

Одна из книг "Рефакторинг. Улучшение существующего кода": Мартин Фаулер с соавторами пролили свет на процесс рефакторинга, описав принципы и лучшие приемы его осуществления, а также указав, где и когда следует начинать углубленное изучение кода с целью его улучшения.

8. Грейс Хоппер (англ. Grace Hopper, урожденная Grace Brewster Murray — Грейс Брюстер Мюррей; 9 декабря 1906 — 1 января 1992) — американский военный деятель, контр-адмирал, программист, создала программное обеспечение для компьютера Марк I и, предположительно, разработала первый в истории компилятор для языка программирования. В её честь был назван эсминец ВМФ США USS Hopper (DDG-70).

Родилась 9 декабря 1906 г. в семье страхового агента Уолтера Флетчера Мюррей (Walter Fletcher Murray) в г. Нью-Йорк. В 1923 г. подавала документы для поступления в колледж Вассар (Vassar College), но не прошла экзамен по латыни. Следующая попытка завершилась успешно, и в 1924 г. Грейс Хоппер стала студенткой этого учебного заведения.

"Корабль в порту находится в безопасности, но корабли строят не для этого. Они должны плавать и делать что-то новое." Хоппер Грейс.

9. Эдсгер Вибе Дейкстра (нидерл. Edsger Wybe Dijkstra; 11 мая 1930, Роттердам (Нидерланды) — 6 августа 2002) — выдающийся нидерландский учёный, идеи которого оказали огромное влияние на развитие компьютерной индустрии. Родился 11 мая 1930 года в Роттердаме, в семье учёных (отец — химик, мать — математик).

По окончании школы поступил на факультет теоретической физики Лейденского университета. В 1951 году увлёкся программированием, поступил на трёхнедельные компьютерные курсы в Кембридже, с 1952 года работал программистом в Математическом центре Амстердама под руководством профессора Ван Вейнгаардена (впоследствии — автора одного из способов формального описания грамматики формальных языков — так называемых двухуровневых грамматик Ван Вейнгаардена).

"Если отладка — процесс удаления ошибок, то программирование должно быть процессом их внесения." Дейкстра Эдсгер Вибе

Уже в 1952 году принял решение окончательно специализироваться на

программировании, но курс теоретической физики закончил. В 1956 году принял участие в разработке ЭВМ X1. Эта машина была создана тремя энтузиастами за год. Именно для оптимизации разводки плат для X1 был придуман алгоритм поиска кратчайшего пути на графе, известный как «алгоритм Дейкстры».

10. Академик **Андрей Петрович Ершов** (1931-1988) - один из зачинателей теоретического и системного программирования, создатель Сибирской школы информатики. Его существенный вклад в становление информатики как новой отрасли науки и нового феномена общественной жизни широко признан в нашей стране и за рубежом.

Еще студентом МГУ, под влиянием А. А. Ляпунова он увлекся программированием. Закончив университет, А. П. Ершов поступил на работу в Институт точной механики и вычислительной техники - организацию, в которой складывался один из первых советских коллективов программистов.

В 1957 г. его назначают заведующим отделом автоматизации программирования во вновь созданном Вычислительном центре АН СССР. В связи с образованием Сибирского отделения АН СССР по просьбе директора Института математики СО АН СССР академика С. Л. Соболева он берет на себя обязанность организатора и фактического руководителя отдела программирования этого института, а затем переходит в Вычислительный центр СО РАН.

11. Датой рождения **Билла Гейтса** является 28 октября 1955 года. Он родился в семье Вильяма Гейтса, корпоративного адвоката, и Мэри Максвелл Гейтс, члена совета директоров First Interstate Bank.

Билл Гейтс учился в самой привилегированной школе Сиэтла. Родители ожидали, что он пойдет по стопам отца и поступит в Гарвардскую школу права. Однако Гейтс не преуспевал в грамматике, граждановедении и других предметах, которые он считал тривиальными, к седьмому классу он увлекся математикой и мечтал стать профессором. В 1968 году, когда Билл и его школьный товарищ [Пол Аллен](#) учились в средних классах, руководство школы решило купить компьютерное время у компании General Electric. В то время рынком правили системы на основе микроархитектуры DEC PDP-10.

Это изменило жизнь Билла. Он и Аллен увлеклись не на шутку, они даже прогуливали занятия, чтобы изучить всю доступную компьютерную литературу.

12. **Стивен Пол Джобс (Steven Paul Jobs)**, американский изобретатель, предприниматель и сооснователь корпорации Apple, родился 24 февраля 1955 года в Сан-Франциско, штат Калифорния.

Абдулфаттах Джандали и Джоан Симпсон, биологические родители Стива, отдали ребенка на усыновление и он вырос в приемной семье, где его родителями стали Пол и Клара Джобс.

В 1972 году Стив Джобс поступил в Рид-колледж в Портленде, штат Орегон, где был отчислен после первого семестра, но продолжал посещать ряд занятий, среди которых была каллиграфия. В последствии Стив утверждал, что если бы он не прошел курс по каллиграфии, то «у Мака никогда бы не было нескольких гарнитур или пропорциональных шрифтов».

В 1974 году Джобс вернулся в Калифорнию и устроился на работу в компанию Atari, занимавшуюся производством компьютерных игр.

5 Подведение итогов

6 **Домашнее задание.** Создать логическую структуру сайта по выбранное тематике

7. Критерии оценки деятельности учащихся

Зачет	Выполнена лабораторная работа. Выполнен верный вариант (при задании вариантов). Выполнен отчет по лабораторной работе с указанными в задании пунктами. Учащийся демонстрирует понимание выполненной работы. Учащийся может выполнить аналогичное задание.
Незачет	Один или более пунктов указанных выше не выполнен или выполнен не верно.

Разработал преподаватель:

З.В.Зенько

2016-2017