

## **Содержание:**

### **Введение**

Все в большей степени успешная практическая деятельность современного целеустремленного человека зависит от эффективной организации обмена информацией. Увеличились информационные потоки и повысились требования к скорости обработки, передачи данных, а так же своевременному владению информацией, ее усваиванию и скорости ее познания.

В связи с этим развитием информационных технологий, сетей, а также информационных систем получили широкое распространение гипертекстовая технология, языки гипертекстовой разметки, в частности HTML. Информационные системы при этом рассматриваются как инструмент моделирования реальности, реализующей различные подходы.

В последние годы стали появляться инструментальные средства и крупные информационные системы, в которых совместно используются различные информационные технологии.

Курсовая работа предназначена для обобщения накопленного отечественного и зарубежного опыта в разработке информационных систем связанная с интернет-технологиями, выявление общих положений и принципов их построения и развития.

Цель курсовой работы - изучить теоретический материал по тематике курсовой работы.

Для достижения поставленной цели были выделены следующие задачи:

Во-первых, проанализировать литературу по теме курсовой работы;

Во-вторых, рассмотреть и изучить понятия: «гипертекстовая технология»

В-третьих, охарактеризовать основные технологии Web, такие как: HTML, XML, XHTML;

# Глава 1. Краткая характеристика языков гипертекстовой разметки

Основой современных технологий, используемых в Internet, являются языки разметки гипертекста и написания сценариев, апплетов и скриптов публикаций WWW (World Wide Web). Сегодня разработано достаточно много различных программных языков, либо специально ориентированных на использование в Web, либо включающих функции, предназначенные для Internet.

В 1986 г. Международная организация по стандартизации (ISO) приняла первый стандарт ISO-8879, озаглавленный «Standard Generalized Markup Language (SGML)». Этот стандарт был посвящен описанию языка SGML - обобщенного метаязыка, позволяющего строить системы логической и структурной разметки любых разновидностей текстов.

Он был разработан для стандартизации обмена данными между разнородными программами и платформами. Язык SGML являлся не только мощным, но и очень сложным средством. Кроме того, SGML-документ было трудно интерпретировать без определений языка разметки, который хранится в определении типа документа DTD (document type definition). DTD необходимо было посылать вместе с SGML-документом или включать в документ, для того чтобы можно было распознать теги, созданные пользователем. Под тегами понимались специальные управляющие коды, записываемые в виде текста, ограниченного угловыми скобками

Необходимо отметить, что основной проблемой того времени оставалась проблема корректного отображения ранее созданных страниц текста ввиду достаточно большого разнообразия программ создания текстовых документов и различных аппаратно-программных платформ.

В связи с этим разработчики SGML стремились создать программно- и аппаратно-независимые языки, с тем чтобы размеченный текст мог без труда интерпретироваться любой программой, работающей с любым устройством вывода.

Фактически SGML представлял собой не готовую систему разметки текста, а лишь удобный метаязык, позволяющий строить такие системы для конкретных обстоятельств. Стандарт SGML определял лишь синтаксис записи элементов разметки текстов - тегов и их атрибутов, а также содержал правила определения

новых тегов и указания структурных отношений между ними. Для практической же разметки документов использовалось приложение SGML: набор определенных в соответствии со стандартом тегов, являющийся, по сути, формальным описанием структуры документа. Теги представляли собой последовательность символов, обрамляющих текст.

Однако язык SGML не получил массового распространения до тех пор, пока он не был выбран в качестве основы для нового языка разметки гипертекстовых документов. Этот язык (самое известное из приложений SGML) был назван HTML (Hyper Text Markup Language, «язык разметки гипертекста»).

По сути, язык HTML - это набор предписаний SGML, сформулированных в виде DTD. В случае языка HTML DTD хранится в браузере и этот DTD во много раз меньше, чем DTD SGML. Для передачи информации по Internet HTML использует протокол передачи гипертекстов (Hypertext Transfer Protocol).

HTML - это технология представления данных, не несущая содержательной информации, кроме заключенной в тегах. Значимость тегов в этом языке не определена, поэтому нельзя представить иерархию данных. В качестве платформы для приложений используются браузеры.

HTML был попыткой разрешить уже достаточно очевидное к тому времени противоречие между идеологией структурной разметки и потребностями пользователей, заинтересованных в первую очередь в гибких и разнообразных возможностях визуального представления информации.

Чтобы разрешить это противоречие, не нарушая основные концепции разметки гипертекста, авторы версии 3.0 ввели новое средство: иерархические стилевые спецификации (Cascading Style Sheets - CSS). Система CSS формально независима от HTML, имеет совершенно иной синтаксис, не наследует никаких идеологических ограничений и позволяет уже совершенно в иных терминах задавать параметры визуального представления для любого тега HTML. С помощью CSS автор может указать, каким шрифтом, какого кегля набирать заголовки такого-то уровня. Причем это было достигнуто не введением новых тегов, а разработкой механизма, позволяющего влиять на интерпретацию уже существующих тегов логической разметки, с учетом контекста его употребления. Браузеры с текстовым интерфейсом или синтезаторы речи могут проигнорировать эту дополнительную информацию и использовать для оформления свои обычные средства.

В поддержку языка HTML была создана целая серия программных языков, таких, как Java, JavaScript, Perl, PHP и т.д. Тем самым открылась возможность для создания динамических Web-страниц, включающих формы, таблицы, фреймы и другие элементы.

Упростив SGML для использования в Web, группа предложила расширяемый язык разметки XML (Extensible Markup Language - рекомендация W3C по статусу на февраль 1998 г.). XML представляет собой подмножество SGML, причем любой действительный документ XML является действительным документом SGML[Оригинал статьи The future of HTML, Part 2: XHTML 2.0.

([www.ibm.com/developerworks/web/library/x-futhtml2.html](http://www.ibm.com/developerworks/web/library/x-futhtml2.html))]. Как и SGML, XML - это метаязык, определяющий другие языки разметки для специфических целей. Например, язык синхронизированной интеграции мультимедиа (Synchronized Multimedia Integration Language - SMIL) базируется на XML.

Чтобы сделать Web-страницу интерактивной, необходим язык, выполняемый в контексте браузера. В качестве таких языков выступают Java, JavaScript, PHP, Perl и др.

Создаваемые программы, называемые сценариями или скриптами, включаются в состав Web-страниц и распознаются и обрабатываются браузером отдельно от остального HTML-кода.

Обычно скриптовые языки используются для:

отображения диалоговых панелей и сообщений в статусной строке браузера;

создания динамического содержимого страницы во время ее загрузки или уже после того, как она полностью загружена;

изменения содержимого страницы, атрибутов графического изображения, содержимого других HTML-элементов и отклика на события;

проверки вводимой пользователем информации и управления отсылкой содержимого форм;

навигации по другим страницам;

управления встроенными в Web-страницу объектами типа Java-апплетов и элементов ActiveX;

обеспечения связи Web-страницы с базами данных.

## Глава 2. Языки гипертекстовой разметки

### 2.1 Язык разметки гипертекста HTML

Электронные версии печатных изданий на базе WWW подготавливаются, хранятся и используются в среде World Wide Web. WEB-публикации представляют собой многосложные гипертекстовые структуры, для создания которых был разработан специальный язык, получивший название «язык разметки гипертекста» (HTML).

**Язык разметки гипертекста** (от англ. HyperText Markup Language, сокращённо **HTML**) - это стандартный язык разметки веб-документов, который определяет, как и какие именно элементы должны будут располагаться на веб-странице. Многие считают, что HTML это язык программирования, но это не так, он не может выполнять простые действия (сложение, вычитание и т.п.), кроме того, в HTML не входят основные элементы всех языков программирования (функции, циклы, переменные и т.п.).

В основном веб-страницы создаются при помощи языка HTML или XHTML (от англ. Extensible Hypertext Markup Language - расширяемый язык разметки гипертекста). Язык HTML интерпретируется браузером и отображается в виде документа, в удобной для человека форме восприятия.

Язык HTML был разработан британским учёным [Тимом Бернерсом-Ли](#) приблизительно в 1989-1991 гг. в стенах Европейского совета по ядерным исследованиям в Женеве (Швейцария). HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. Таким образом с помощью HTML можно легко создать относительно простой, но красиво оформленный документ.

Текст с разметкой HTML должен по идее одинаково воспроизводиться на оборудовании с различной технической оснащённостью (монитор, планшет, мобильный телефон, и тп) и программным обеспечением, но на практике не всегда так. Потому что существует много различных браузерных движков (Presto, WebKit, Gecko, Trident), которые иногда отходят от стандартов или интерпретируют их по своему (к примеру Internet Explorer, для которого происходит "доделывание" веб-

страницы под все его версии).

Современное применение HTML очень далеко от его изначальной задачи. К примеру, тег **<table>**, предназначен для создания в документах обычных таблиц, но долгое время использовался для верстки веб-страниц, где таблицы с невидимой границей, позволяли разделять документ на модульные блоки. Теперь же ему на смену пришел более современный способ верстки с помощью слоев. Хотя и табличная верстка до сих пор популярна и используется.

С течением времени, основная идея платформонезависимости языка HTML была отдана в своеобразную жертву современным потребностям в мультимедийном и графическом оформлении. **HTML** - теговый язык разметки документов. Любой документ на языке HTML представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными пометками - тегами. Элементы могут быть пустыми, то есть не содержащими никакого текста и других данных (например, тег перевода строки **<br>**). В этом случае обычно не указывается закрывающий тег. Кроме того, элементы могут иметь атрибуты, определяющие какие-либо их свойства (например, выравнивание содержимого контейнера **<div>** по краю). Атрибуты указываются в открывающем теге.

Язык HTML тесно взаимодействует с такими программными разработками, как стили CSS и Javascript и для чего необходимо такое сотрудничество языков.

Язык программирования HTML5 (от английского HyperText Markup Language) - это пятая версия языка. Как и все предыдущие версии, предназначен для выстраивания структуры документов и представления их в сети. На данный момент версия HTML5 находится в стадии разработки. Цель, преследуемая созданием пятой версии HTML - улучшение языка в области его работы с документами мультимедиа (аудио- и видеоприложениями).

Добавление некоторых синтаксических новшеств, таких, как **<audio>**, **<video>** и **<canvas>**, позволит минимизировать логические и синтаксические ошибки при работе HTML5 с мультимедийными файлами. Также для обогащения семантического содержимого документа (семантика - изучение значения языковых единиц) вводятся и новые значения, такие, как **<article>**, **<section>**, **<nav>** и **<header>**. Некоторые из атрибутов будут удалены или изменены. Вводятся новые возможности для обработки недопустимых документов, так что все браузеры будут способны работать с такого рода документами.

Спецификация HTML определяет набор символов для документов HTML как набор однобайтовых символов ISO 8859/1 (латинский алфавит - Latin 1). Это набор из 256 символов, из которых первые 128 идентичны символам ASCII (ISO-646). Они должны быть совместимым с ISO 10646 и Unicode.

## 2.2 XHTML (англ. Extensible Hypertext Markup Language)

**XHTML**. Расшифровывается как **Extensible Hypertext Markup Language** и на русском - Расширяемый язык разметки гипертекста. Обратите внимание не расширенный, а расширяемый. Это значит, что данный язык пополняется (расширяется) до сих пор.

Основным отличием XHTML от HTML является способ обработки документа (интернет-странички). Стоит ввести еще одно определение. **Парсер** — программа или часть программы, которая выполняет синтаксический анализ. Также его еще называют **синтаксический анализатор**. Если еще проще, то данная программа выполняет анализ всего строения страницы, всего кода страницы. В HTML при нахождении ошибки, во время анализа она исправлялась, что требовало дополнительного времени — браузеру нужно было понять, что автор (разработчик) хотел написать. В XHTML эти ошибки уже не исправлялись. Например, при ошибке в каком-либо теге, он просто выводился с остальным текстом.

Еще одно отличие заключается в том, что все элементы должны быть закрыты, а одиночные теги должны после символов иметь знак /, например: `<br />`. Про тэги я напишу отдельную большую статью, также напишу про каждый тэг отдельно. Кодировка в XHTML — UTF-8 (наиболее распространенная сейчас), в HTML же использовалась ISO 8859-1.

Версий XHTML меньше, чем HTML.

Версия 1.0 была разработана и принята в 2000 году, вторая редакция этой же версии вышла двумя годами позднее;

Версия 1.1 отличалась тем, что авторы и разработчики смогли добавлять свои элементы в разметку;

Затем вышли две мобильные версии — XHTML Basic и XHTML MobileProfile (MP). Облегченные версии XHTML 1.0, разработанные специально для мобильных устройств;

И, наконец, последняя версия — 2.0. Но ее как таковой не существует, т.к. разработка остановилась в конце 2010 года, а все что успели сделать передали разработчикам HTML 5.

## 2.3 Язык Java

Язык программирования Java (изначально “Oak” - дуб) был создан группой сотрудников “Sun Microsystems” и первоначально предназначался для программирования бытовой электроники. Однако, добиться широкого применения в тостерах и кофемашинах им так и не удалось. В то же самое время интернет начал развиваться семимильными шагами, поэтому разработчики, по велению руководства компании, переключились на работу с вебом. Итогом их работы стал мощный интерактивный браузер, превзошедший во всех отношениях популярный тогда “Mosaic”, написанный на C. Он позволял выполнять код на странице, например, создавать примитивную графику или взаимодействовать с пользователем, а также мог работать на любом устройстве. Это была настоящая сенсация в сфере IT, после чего Java надолго стала доминирующим языком программирования.

В итоге, авторы решили сменить название на нечто более близкое по духу для программистов. По одной из версий, Java была названа в честь марки кофе с острова Цейлон (Ява), который создатели языка, видимо, пили утром после очередной бессонной ночи кодинга.

### Философия Java

Основной задачей Java было то, чтобы написанный программистом код можно было бы без изменений запускать на любом устройстве. Сейчас в это, возможно, трудно поверить, но даже в 90-ых одну и ту же программу для работы на компьютерах с другой архитектурой приходилось переделывать едва ли не полностью. Чтобы этого избежать, в Java была придумана следующая концепция - всё, что написал программист переводится компилятором в байт код, который не зависит от платформы. Этот байт код выполняется при помощи интерпретатора, называемого JVM (виртуальная машина Java). Виртуальная машина должна быть установлена на

устройство заранее и зависит от платформы. Существует множество готовых версий JVM, которые требуется только установить и скачать. Поэтому программисту на Java не нужно беспокоиться о совместимости его программы с архитектурой того или иного компьютера.

Текущей версией является Java 10, которая вышла в марте 2018 года. А Java превратилась из просто универсального языка в целую платформу и экосистему, которая объединяет различные технологии, используемые в целого ряда задач: от создания десктопных приложений до написания крупных веб-порталов и сервисов. Кроме того, язык Java активно применяется для создания программного обеспечения для целого ряда устройств: обычных ПК, планшетов, смартфонов и мобильных телефонов и даже бытовой техники. Достаточно вспомнить популярность мобильной ОС Android, большинство программ для которой пишутся именно на Java [9].

### Особенности Java

Ключевой особенностью языка Java является то, что его код сначала транслируется в специальный байт-код, независимый от платформы. А затем этот байт-код выполняется виртуальной машиной JVM (Java Virtual Machine). В этом плане Java отличается от стандартных интерпретируемых языков как PHP или Perl, код которых сразу же выполняется интерпретатором. В то же время Java не является и чисто компилируемым языком, как C или C++.

Подобная архитектура обеспечивает кроссплатформенность и аппаратную переносимость программ на Java, благодаря чему подобные программы без перекомпиляции могут выполняться на различных платформах - Windows, Linux, Mac OS и т.д. Для каждой из платформ может быть своя реализация виртуальной машины JVM, но каждая из них может выполнять один и тот же код.

Java является языком с Си-подобным синтаксисом и близок в этом отношении к C/C++ и C#. Поэтому, если вы знакомы с одним из этих языков, то овладеть Java будет легче.

Еще одной ключевой особенностью Java является то, что она поддерживает автоматическую сборку мусора. А это значит, что вам не надо освобождать вручную память от ранее использовавшихся объектов, как в C++, так как сборщик мусора это сделает автоматически за вас.

В Java, как и во многих современных языках, включая C++, Python, etc, используется принцип объектно-ориентированного программирования (ООП). Java в основном используется для создания серверных приложений и мобильных ПО. Также это основа нативных приложений под Android. Данный язык чрезвычайно популярен среди разработчиков, хоть и есть вероятность, что в плане мобильных приложений всю популярность вскоре может отобрать Kotlin.

Где применяется Java

Спектр применения Java довольно широк. Стоит упомянуть что существует некоторое разделение языка по сферами его применения. Это Java SE (стандартная версия) и Java EE (версия для бизнес-приложений).

Первая - базовый пакет, предназначенный, в основном, для создания десктопных программ. Например, на Java SE Вы можете написать свой аналог Paint или игру для ПК, а также апплеты для взаимодействия с веб.

Java EE является надстройкой над SE и предназначена для создания более солидных приложений, тесно связанных с вебом. В нее добавлены дополнительные библиотеки для работы с сетью, БД и все что нужно для клиент-серверного взаимодействия. Например, если когда-нибудь Вам понадобится создавать банковскую систему, которая будет обрабатывать множество транзакций, хранить и выдавать информацию о состоянии вкладов клиентов в режиме онлайн или же свою социальную сеть, то Вам нужно будет использовать версию EE.

Стоит упомянуть также и Java ME(Micro Edition), предназначенную для мобильных устройств. Это более укороченная версия Java SE со специальными библиотеками функций, для поддержки смартфонов, планшетов и других портативных устройств. На ней также разрабатывают софт для ресиверов цифрового телевидения и проигрывателей Blu-ray. Кстати, как это ни парадоксально, но разработка под Android больше относится к SE, чем к Micro Edition.

Примеры сервисов и программ, созданных с использованием Java, не нуждающихся в представлении:

Amazon;

Ebay;

Yahoo;

Юлмарт

Одноклассники

Minecraft

торрент-клиент Vuze (а также многие другие торренты и менеджеры загрузок)

Плюсы/минусы Java

Некоторые преимущества Java были описаны выше - это кроссплатформенность, универсальность (работает и с вебом, и с десктопом). Ещё для Java существует огромное количество open source библиотек и программ, которые можно использовать в своих проектах.

Из минусов Java стоит отметить, пожалуй, самую известную проблему - это потребление памяти. Например, написанная на Java система багтрекинга Jira может использовать столько же ресурсов, сколько тратит какой-нибудь мейнфрейм для анализа бозона Хиггса в Большом Адронном Коллайдере, причем даже в те моменты, когда не совершает каких то сложных операций. Это, разумеется, шутка, хотя в каждой шутке есть доля правды. Но проблемы с памятью, чаще всего, решаются оптимизацией кода, к тому же язык тоже постоянно оптимизируется и совершенствуется, поэтому за последние годы проблемы с утечкой памяти в Java существенно снизились.

Сопутствующие технологии

**Java FX** - платформа, которая чаще всего используется для программ работающих с анимацией и мультимедиа, например при создании игр.

**JSP** - позволяет создавать веб-странички одновременно со статичными элементами (HTML, SVG) и динамичными.

**Swing** - библиотека Java, которая позволяет создавать программы с графическим интерфейсом. Сюда относятся кнопки, поля ввода, меню, чекбоксы и так далее.

**Groovy** - java-подобный язык, который использует упрощенный синтаксис Java. Groovy обладает некоторыми возможностями, недоступными в Java и может использоваться в любом java проекте. Например, для написания тестов.

**Hibernate** - библиотека Java, которую используют для работы с реляционными базами данных. С её помощью можно, например, автоматизировать генерацию SQL

запросов. Hibernate можно использовать как для создания базы данных с нуля, так и для работы с уже существующими БД.

**Spring** - один из самых известных фреймворков для Java. Используется для создания сложных корпоративных приложений и сильно упрощает для программиста работу с JavaEE.

## 2.4 Язык JavaScript

При написании веб-приложений, программирование на JavaScript используется наиболее часто. Если кратко перечислить ключевые особенности данного языка, то следует выделить следующее:

Объектно-ориентированность. Выполнение программы представляет собой взаимодействие объектов;

Приведение типов данных проводится автоматически;

Функции выступают объектами базового класса. Эта особенность делает JavaScript похожим на многие функциональные языки программирования, такие как Lisp и Haskell;

Автоматическая очистка памяти. Так называемая, сборка мусора делает JavaScript похожим на C# или Java.

Если говорить о сути применения JavaScript, то этот язык позволяет «оживлять» неподвижные страницы сайтов с помощью кода, который можно запустить на исполнение (так называемые, скрипты). То есть, можно провести аналогию с мультфильмами, где html и css- это прорисованные герои, а JavaScript – это то, что заставляет их двигаться.

Если говорить о синтаксисе JavaScript, то ему присущи следующие особенности:

Регистр важен. Функции с названиями func() и Func() – совершенно разные;

После операторов необходимо ставить точку с запятой;

Встроенные объекты и операции;

Пробелы не учитываются. Можно использовать сколько угодно отступов, а также переводов строки, чтобы оформить свой код.

Простейший код на JavaScript выглядит следующим образом:

## Сфера применения

Для того чтобы понять, зачем нужен JavaScript, и насколько необходимо его изучение, следует выделить некоторые области, в которых применяется данный язык программирования.

Разработка веб-приложений. Хотите установить простой счетчик, организовать передачу данных между формами или поместить на своем сайте игру? Тогда JavaScript выступит верным помощником в этом деле;

«Активное участие» в AJAX. Эта технология позволила значительно ускорить работу приложений, осуществляя обмен данными с сервером в «фоновом» режиме:

Операционные системы. Возможно, кто-то не знал, но Windows, Linux и Mac имеют своих браузерных конкурентов, львиная доля кода которых написана на JavaScript;

Мобильные приложения;

Сфера обучения. Любая программистская специальность в университете включает в себя изучение JavaScript в том или ином объеме. Это обусловлено тем, что язык изначально разрабатывался для не очень сильных программистов. Уроки JavaScript логически вплетаются в базовый курс HTML, поэтому освоение проходит достаточно просто.

## Преимущества и недостатки

Не стоит думать, что JavaScript – это какая-то панацея от всех проблем, и каждый программист с улыбкой на лице пользуется этим языком. Всё на свете имеет свои положительные и отрицательные стороны. Для начала, отметим недостатки.

Необходимость обеспечивать кроссбраузерность. Раз уж JavaScript выступает как интернет-технология, то приходится мириться с правилами, которые устанавливает всемирная паутина. Код должен корректно выполняться во всех, или хотя бы самых популярных, браузерах;

Система наследования в языке вызывает трудности в понимании происходящего. В JavaScript реализовано наследование, основанное на прототипах. Люди, изучавшие другие объектно-ориентированные языки программирования, привыкли к привычному «класс потомок наследует родительский класс». Но в JavaScript такими

вещами занимаются непосредственно объекты, а это не укладывается в голове;

Отсутствует стандартная библиотека. JavaScript не предоставляет никаких возможностей для работы с файлами, потоками ввода-вывода и прочими полезными вещами;

Синтаксис в целом затрудняет понимание. Красота кода – явно не конёк JavaScript, но главное правило программистов соблюдено: «Работает? Не трожь!».

Теперь стоит отметить некоторые преимущества

JavaScript предоставляет большое количество возможностей для решения самых разнообразных задач. Гибкость языка позволяет использовать множество шаблонов программирования применительно к конкретным условиям. Изобретательный ум получит настоящее удовольствие;

Популярность JavaScript открывает перед программистом немалое количество готовых библиотек, которые позволяют значительно упростить написание кода и нивелировать несовершенства синтаксиса;

Применение во многих областях. Широкие возможности JavaScript дают программистам шанс попробовать себя в качестве разработчика самых разнообразных приложений, а это, безусловно, подогревает интерес к профессиональной деятельности.

## **2.5 PHP - язык написания сценариев**

PHP(Personal Home Pages Tools) был создан программистом из Дании, Расмусом Ледорфом, как набор скриптов, написанных на Perl, для учета посетителей его личной странички (онлайн-резюме) в 1995 году. Вскоре после этого Расмусу понадобилось расширить возможности этого набора, поэтому он написал на C новый интерпретатор, который назвал PHP/FI. Язык унаследовал от Perl стиль обозначения переменных, но синтаксис был намного проще. Через два года автор языка выпустил вторую версию.

PHP/FI 2.0 не пользовался большой популярностью, пока в 1997 году программисты из израильского города Хайфа, Зеев Сураски и Энди Гутманс, не предложили Расмусу Ледорфу вместе работать над улучшением PHP. В итоге, весь код был полностью изменен и через некоторое время выпустили PHP 3.0 (от аббревиатуры

FI решили отказаться). В результате этих изменений язык стал очень популярным и стал использоваться на огромном количестве веб-сайтов. Для большей солидности аббревиатуру поменяли на “PHP: Hypertext Preprocessor”, чтобы язык не ассоциировался только с домашними страничками.

Вскоре после выпуска третьей версии Сураски и Гутманс начали работу над PHP 4.0. Новый движок для языка назывался Zend Engine, а их фирма, которую они впоследствии основали, стала называться Zend Technologies. Пятая версия была представлена в 2004 году, в ней была улучшена производительность и изменены функции ООП (взяли за основу модель из Java).

PHP 6.0 так и не вышла (из-за проблем с Юникодом), после чего долгое время развивалась только пятая версия. Однако, в 2014 году начали разработку PHP 7.0, которая вышла в конце 2015.

### Философия PHP

PHP очень сильно похож на C, однако в нем есть некоторые заимствования из Perl( например, обозначение переменных начинается со знака доллара). Как и в JavaScript, PHP код разрешено включать прямо в html-страницы или же создавать отдельные файлы, но в PHP они будут выполняться на стороне сервера (JS скрипты можно запускать локально в браузере, а для PHP нужен web-сервер ).

Начиная с версии 3.0, в PHP была добавлена возможность написания своими силами дополнительных модулей, что немало помогло популяризации и распространению языка. Также стоит отметить, что ядро языка автоматически управляет памятью, возвращая её системе после выполнения скрипта, поэтому программисту не нужно заботиться об этом.

### Где применяется PHP

PHP один из наиболее популярных языков в Web, на нем написано огромное количество сайтов, блогов, фотогаллерей, интернет-магазинов, сайтов знакомств, форумов, социальных сетей (ВКонтакте и Facebook, хотя сейчас они используют свои версии языка, KPHP и HPHP ). Движок для Википедии (MediaWiki) написан на PHP.

На PHP редко приходится делать что-то с нуля потому, что уже существует большое количество готовых решений и систем. Для создания новостных сайтов или аналогичных есть три CMS (систем управления содержимым) о которых,

наверное, слышали все - это Joomla, Wordpress и Drupal. Для интернет магазинов есть системы управления OpenCart, Magento. Все они бесплатны, за исключением дополнительных модулей, которые не входят в них.

Для создания GUI-приложений PHP применяется редко, однако всё же есть такие расширения и библиотеки, как PHP-GTK и PHP-Qt, а также некоторые IDE и пакеты (WinBinder), при помощи которых можно создать оконное приложение.

## 2.6 Язык Perl

В переводе на русский язык аббревиатура Perl звучит как «практический язык, чтобы извлекать данные и составлять отчеты». Символом этого вида языка решили сделать верблюда, наверное, посчитав, что, так как животное это благородное, выносливое, способно много дней безропотно нести тяжелую ношу, такой же труд напоминает нагрузку на новый язык, тоже нешуточную. Почему новый язык сразу оценили многие пользователи, есть много причин. Кто пользуется системой UNIX, знает, что настройки здесь проходят через специальные файлы - обычные текстовые, а изменения осуществляются после выполнения команд. Они написаны на особом языке оболочки, а их выполнение происходит из командной строки. Система UNIX позволяет создавать индивидуальные команды, опираясь на команды интерпретатора и сохраняя их. Они остаются в текстовых файлах, и если требуется, выполняются как стандартный набор команд операционной системы - только через командную строку. Команды пользователя иначе именуется скрипты.

В системе UNIX приходится операционному администратору писать большое количество скриптов, которые, в свою очередь, обрабатывают последующие скрипты, т. е. текстовые файлы с использованием специальных программ: Awk. Программа дает возможность сопоставлять с данными образцами и служит генератором отчетов. Sed. Выступает как пакетный редактор для текстовых файлов. Программы в качестве фильтров поочередно считывают строки входных файлов. Затем они выполняют применимые к строке действия, которые определяются благодаря командам программ. При этом выделяется цепочка символов по определенным шаблонам, замена по установленным правилам, генерируются новые файлы.

Новый язык программирования Perl открыл очень богатые возможности в тех случаях, когда требуется обработать текст, устойчивые выражения. Отличает Perl большие варианты вспомогательных модулей. Со временем этот уникальный язык

расширил свои границы, и теперь широко используется в веб-разработках и играх, если необходимо сетевое программирование или графическая разработка интерфейсов для пользователей. Язык прижился и полюбился за легкость использования, сейчас его признают как самый динамический язык программирования. Он поддерживает разнообразные парадигмы, куда входят стили – функциональный, процедурный и другие, у него имеется контроль за памятью, есть поддержка обработки текста. Не зря один из девизов Л. Уолла гласит, что вещи простые могут оставаться простыми, но сложные обязаны выполняться. Освоить язык несложно, разобравшись в выражениях присваивания, блоках кода, управляющих структурах и функциях. Схожесть с другими языками Ларри многое позаимствовал из других языков программирования. Например:

Командные оболочки UNIX. Переменные маркированы ведущими знаками, они четко выражают тип переменной и помогают этим переменным являться в строках интерполированными. Встроенные функции обеспечивают инструментарий, а он используется для программирования оболочки.

Массивы из Лиспа. Использование Perl регулярных выражений из awk, заодно позаимствовав от него ассоциативные мотивы. Из sed.

Но с собственными задачами новый язык справлялся гораздо быстрее, и его, поэтому начали применять не только для решения задач системного администрирования. Уолл почерпнул у Г. Спенсера пакет, который помог освоить Perl регулярные выражения, модифицируя его под свое детище. Многие разработанные функции – заслуга не только Ларри. Коллеги и друзья не менее увлеченно старались привнести в разрабатываемую уникальную программу свои новшества. А когда язык появился в интернете, образовалось целое сообщество единомышленников, которые тоже помогли его усовершенствовать.

В работе задействовано с тех пор, как утверждает статистика, более 10 000 программистов. Впрочем, язык и сейчас усовершенствуется благодаря разработке пакетов, которые используются с успехом в реализации новых применений языка под развивающиеся технологии, связанные с информацией.

Perl 5 активно разрабатывается и теперь, пополняясь все новыми версиями. Но все разработки роднит несколько факторов: выполняется автоматическая типизация данных, проводится автоматический контроль над памятью. Благодаря интерпретатору известны не только типы, но и запросы памяти каждого объекта программы, и ему не составляет труда распределить память, освободить при

подсчете ссылок. Нужно отметить, что Perl-программирование отличает интерпретатор от других. Доказательством тому являются трансляция программы в промежуточный байт-код, а уже потом начинает выполняться. В сравнение: традиционный работает немного иначе – вводимый оператор интерпретируется и мгновенно выполняется, что чревато синтаксическими ошибками во время рабочего процесса. Perl позволяет обнаружить такого рода ошибки уже в период трансляции в бай-код.

Двухтысячный год подарил миру новую, 5.6, версию языка перл. Она уже выглядела намного солидней первых, имела 64-битные системы, в состоянии была поддерживать файлы более двух Гб, включала представление строк, учитывая стандарт юникод, а также ключевое слов – `our`. В этот период меняется схема наименований версий, чтобы она была более близкой к другим проектам – с открытым исходным кодом. Интересным фактом является такое решение: если в разработке находится версия, ей будет присвоен номер с нечетными цифрами, а стабильная получает четную нумерацию. Уолл тогда обратился к своим приверженцам с просьбой вносить предложения для создания более продвинутой версии. Его призыв был услышан, а итогом явился 361 документ, что помогло разрабатывать VI версию. На суд пользователям представили документ, который, скорее всего, походил на сводку, а эта версия существовала, но лишь в виде описания языка. Ряд документов превратился в спецификацию этой версии. Попытка запустить в действие интерпретатор VI версии в 2006 году была остановлена. Но с 2009 года под названием Rakudo Perl эта версия живет и периодически ее обновляют. Значительное изменение процесса разработки Perl 5 произошло после появления Perl 5.11. Сообщество разработчиков перешло на ежемесячный цикл выпусков, с планированием даты выпуска на три месяца вперед.

Каждый год все новые усовершенствования помогали сделать язык, изобретенный Уоллом, все наиболее легче проще и доступнее. 2010 год ознаменовался тем, что синтаксис языка perl получил поддержку `package NAME VERSION`, перегрузку операторов регулярных выражений, в последующие годы обновлялись модули, расширялась поддержка операторов. Была достигнута цель – указать во время работы ту версию, которую следует эмулировать, значит, позволить обновить perl, но при этом не потерять рабочие способности использованных скриптов, которые представляют угрозу оказаться несовместимыми с представленной новой версией. Стало больше функций, увеличилась безопасность.

В конце мая, появилась очередная версия, как всегда, более усовершенствованная и удобная для работы. Сегодня ее может протестировать любой желающий и на своем лично опыте убедиться, что для создания этого языка программирования проделано немало работы. В этом языке Perl массивы не ограничены. Можно решать, используя язык, нетрадиционные задачи, для осуществления которых было бы в другом случае задействовано несколько, выполнить задания администрирования, обработать качественно текстовые файлы, и на все проблемы времени будет затрачено не так много. Разрабатывая новые программы, используя таблицу символов, задействуя необходимых операторов, результат получится правильный и быстрый.

## 2.7 Язык VRML

VRML: аббревиатура от Virtual Reality Modeling Language (по отечественному ЯМВР, Язык Моделирования Виртуальной Реальности). Стандарт языка описания трехмерных сцен, содержащих объекты, а также возможных взаимодействий между объектами и наблюдателем. Основное назначение — описание набора объектов (сцены) в системах виртуальной реальности. Основное на данный момент применение — представление статической и интерактивной информации в сети интернет, в виде виртуальной трехмерной среды содержащей различные объекты. Существует три распространенных версии языка: VRML 1.0, 2.0 и VRML 97. На данный момент самой широко поддерживаемой версией является VRML 2.0. Развитие языка координируется VRML Architecture (по другим источникам Advisory) Group (перевод — Группа Архитектуры VRML), известной под аббревиатурой VAG.

Язык VRML был задуман весной 1994 г. на первой ежегодной конференции World Wide Web в Женеве. Здесь в числе прочих обсуждались проблемы создания VR-интерфейсов для доступа к WWW и необходимость создания единого общего стандарта описания трехмерной сцены, включающей гиперсвязи по аналогии с HTML. Основными требованиями к языку являлись независимость от платформы, расширяемость и возможность работы по низкоскоростным каналам связи. Там же было предложено название VRML (Virtual Reality Markup Language). Впоследствии слово Markup было заменено словом Modeling, что отражает графическую природу языка. В качестве базы языка VRML был выбран урезанный формат текстового файла SGI Open Inventor, дополненный сетевыми возможностями. Формат Open Inventor позволял описать трехмерную сцену с графическими объектами, источниками света и текстурами..

В язык была добавлена анимация. Ряд объектов, называемых Interpolator, позволял описывать заранее определенное движение, например полет птицы, движение робота или маятника часов. Стало возможным описывать изменение цвета объекта или создавать объекты, изменяющие свою форму. Языку была добавлена интерактивность.

VRML-документ представляет собой обычный текстовый файл, который содержит описания трехмерных фигур и свойств их поверхностей (цвет, текстура материала, освещение и т.п.). Такой документ поступает в браузер в виде исходного текста, точно так же, как и обычный HTML-документ. Браузер, получающий VRML-документ, должен уметь интерпретировать VRML-код. Первая версия языка — VRML 1.0 — была представлена на второй конференции WWW в октябре 1994 года. Стандарт VRML 2.0, принятый в августе 1996 года, поддерживает анимацию и звуковые эффекты; для него существует поддержка на уровне языков Java и JavaScript. В декабре 1997 года ISO и IEC утвердили международный стандарт VRML 97.

Современные версии браузеров обладают встроенной поддержкой языка VRML.

## **Заключение**

Глобальная информатизация общества приводит к тому, что потребность в информации, растет с каждым новым пользователем сети. При этом задачей специалистов в области информационных технологий обеспечить пользователей полной и достоверной информацией путем простого и удобного для пользователей доступа к накопленным массивам данных.

Гипертекстовая технология позволяет быстро и точно осуществить поиск необходимой информации не только в рамках отдельных документов на компьютера и web-страниц, созданных с их использованием, но и в локальных вычислительных сетях и всемирной сети Internet.

Идея гипертекстовой информационной системы состоит в том, что пользователь имеет возможность просматривать документы (страницы текста) в том порядке, в котором ему это больше нравится, а не последовательно, как это принято при чтении книг.

Простой на первый взгляд механизм построения ссылок оказывается довольно сложной задачей, т. к. можно построить статические ссылки, динамические

ссылки, ассоциированные с документом в целом или только с отдельными его частями.

Расширение понятия гипертекста происходит за счет других информационных ресурсов, включая графику, аудио- и видео-информацию, до понятия гипермедиа.

Безусловно у электронных гипертекстовых систем есть множество преимуществ, по сравнению с обычными печатными текстами. Например:

- двунаправленный характер ссылок, по сравнению с направленными только вперед ссылками обычных текстов;
- комментарии к тексту могут быть сделаны в самом тексте, не нарушая в то же время его целостность;
- однажды пройденный маршрут легко запоминается и может быть автоматически пройден многократно; это равносильно тому, что читатель формирует свой личный текст;
- электронная форма представления текста позволяет автоматизировать процесс формирования массива ключевых слов путем автоматического компьютерного анализа текста, и прочие преимущества.

Но при всем многообразии преимуществ прослеживается и ряд недостатков:

- у пользователя теряется ощущение собственного положения и движения в многомерном документе и появляется дополнительная умственная нагрузка для выбора оптимального пути изучения материала;
- чтобы электронная гипертекстовая система была эффективной для пользователя, электронное информационное пространство должно быть достаточно обширным, а применяемые технические средства достаточно мощными;
- создание электронных гипертекстовых систем требует новых подходов к проблеме защиты авторского права: во-первых, это совместный труд большого числа специалистов (в том числе программистов); во-вторых, развитый аппарат электронных гипертекстовых систем обеспечивает возможность непосредственного включения в данную систему текстов других авторов.

При подготовке данной курсовой работы был изучен материал из нескольких книг, в которых поднимались вопросы гипертекстовой технологии, так же были

использованы материалы глобальной сети Интернет.

Материал данной курсовой работы дает необходимые сведения о языках гипертекстовой разметки и пищу для размышлений, о нем можно рассказать еще много интересного и познавательного, для этого следует обратиться к специальным статьям, книгам и прочей информации, посвященной различным аспектам вопросов, посвященным гипертекстовым технологиям.

## Список использованной литературы

1. Брайан Пфаффенбергер, Стивен Шафер, Чак Уайт, Билл Кароу.html, XHTML и CSS. Библия пользователя 3-е изд., 2006 г., 752 с.
2. Дидре Хейз. Освой самостоятельно HTML и XHTML.10 минут на урок.3-е издание. 2015 г., 224 с.
3. Муссиано, Кеннеди.html и XHTML. Подробное руководство, 2015 г., 752 с.
4. Галактионов В.В. Расширяемый язык разметки XML (Extensible Mark-up Language): промышленный стандарт, определяющий архитектуру программных средств Интернет следующего поколения. Дубна, 2016.
5. Оригинал статьи The future of HTML, Part 2: XHTML 2.0. ([www.ibm.com/developerworks/web/library/x-futhtml2.html](http://www.ibm.com/developerworks/web/library/x-futhtml2.html))
6. Справочник по спецификации XHTML 2.0 (<http://www.w3.org/TR/xhtml2>)
7. Гасов В.М. Цыганенко **А.М.** Методы и средства подготовки электронных изданий Учебное пособие <http://www.hi-edu.ru/e-books/xbook081/01/part-008.htm>
8. XHTML 1.0: The Extensible HyperText Markup Language (Открытый язык разметки гипертекстов) Оригинал перевода: <http://www.stack.ru/~julia/XHTML/xhtml1.phtml>
9. Николай Прохоренок - Разработка Web-сайтов с помощью Perl и MySQL, ,Питер , 2015 с.336
10. Итан Браун Web Development with Node and Express: Leveraging the JavaScript Stack ,Питер , 2017 с.336