

Дисциплина: Объектно-ориентированное программирование

Тема 2: Природа и эволюция
сценарных языков

- Сценарные языки, или языки скриптов (scripting languages), за последние годы сделали огромный шаг вперед. Еще лет десять назад им отводилась роль вспомогательных средств, которые и называть-то языками программирования было как-то неловко. Сейчас же скепсис по отношению к ним сменился интересом и признанием. Но какова их природа и какое место они занимают среди всех языков?

Лисп как предтеча сценарных языков

- можно выделить два четких полюса притяжения языков: императивный и декларативный.

Императивные
языки скорее отвечают на
вопрос
«как?»
императивные и родственные
им языки обычно носят
статический характер

декларативные — на вопрос
«что?»
декларативные имеют
динамический характер

Классификация языков

Если составить несколько таблиц, куда будут сгруппированы наиболее значимые и известные языки, которые по тем или иным причинам можно назвать сценарными, то получится четыре класса таких языков:

- командно-сценарные;
- прикладные сценарные;
- языки разметки;
- универсальные сценарные.

Командно-сценарные языки

Название	Год появления	Разработчик языка	Организация, где был создан язык
Pilot	1962	-	IBM
JCL (Job Control Language)	1964	-	IBM
RPG	1965	-	IBM
MUMPS	1969	Окто Барнетт+	Massachusetts General Hospital
sh	1971	Стив Бурн	AT&T Bell Labs
Awk	1977	Альфред Ахо+	AT&T Bell Labs
csh	1978	-	UC Berkeley
Rexx	1979	Майкл Коулишоу	IBM UK Laboratories
AppleScript	1993	-	Apple Computer

Командно-сценарные языки

- Командно-сценарные языки зародились еще в 1960-х гг., когда возникла острая потребность в языках, обеспечивающих управление программами, иначе говоря, языках управления заданиями.
- Наиболее активно из языков этого класса в наши дни используется **Rexx**, созданный в исследовательских лабораториях IBM. По набору средств он мало чем отличается от универсальных сценарных языков, однако выполнен в виде классического блочно-структурированного процедурного языка и предназначен преимущественно для интеграции и расширения функциональности приложений

Прикладные сценарные языки

Название	Год появления	Разработчик языка	Организация, где был создан язык
HyperTalk	1986	-	Apple Computer
Visual Basic	1990	-	Microsoft
JavaScript	1994	-	Netscape Communications, Microsoft
CorelScript	1995	-	Corel
LotusScript	1995	-	Lotus Development
VBScript	1995	-	Microsoft
Pnuts	2001	Тойоказу Томацу	Sun Microsystems

Прикладные сценарные языки

- Прикладные сценарные языки зарождались в 1980-е гг., в эпоху появления промышленных ПК, когда на первый план стали выходить задачи интерактивного общения с ОС, а также доступа к данным электронных таблиц и БД. Отличительная особенность сценарных языков данного класса — ориентация на клиентскую часть ПО.

Прикладные сценарные языки

- Среди прикладных сценарных языков резко выделяется **Visual Basic**, в том числе и такая его разновидность для офисного программирования, как VBA (Visual Basic for Applications). Visual Basic — это тот самый «пограничный» язык, который скорее относится к сценарным, чем к традиционным.
- Несколько особо в этом ряду стоит **JavaScript**, ставший стандартом де-факто в Web- программировании при реализации клиентской части. Его прототипом был язык LiveScript, являвшийся частью серверного продукта LiveWire компании Netscape и первоначально встроенный в Netscape Navigator 2.0. После появления языка Java корпорации Sun Microsystems он начал играть роль самодостаточной надстройки над этим языком, и его название сменилось на JavaScript.
- Из новичков в данном классе языков упомянем экспериментальный язык **Pnuts**, основная идея которого — дать в рамках сценариев наиболее полный доступ к Java API. Его можно использовать для самых разных задач, но прежде всего для оперирования компонентным ПО

Языки разметки

Название	Год появления	Разработчик языка	Организация, где был создан язык
GML	1969	Чарльз Гольдфарб+	IBM
TeX	1979	Дональд Кнут	Stanford University
SGML	1986	-	ISO
HTML	1991	Тим Бернерс-Ли+	CERN
CFML (Cold Fusion)	1995	-	Allaire
DHTML	1996	-	Microsoft, Netscape Communications
XML	1997	-	W3C
XHTML	2001	-	W3C

Языки разметки

- Языки разметки, или тегированные языки, стоят несколько поодаль от линии развития сценарных языков. Их главная отличительная черта — встраивание специального кода (в виде обособленных «команд» — тегов) непосредственно в обычные тексты. Им родственны такие языки, как Postscript и RTF (чаще воспринимаемый просто как особый формат представления документов). Теги стали использоваться для самых разных целей: для отделения структуры информации от ее содержания, для вкрапления команд форматирования и даже для задания динамического поведения встроенных в документ интерактивных объектов.

GML и SGML

- Под влиянием GenCode в 1969 г. американский ученый Чарльз Гольдфарб возглавил работу исследовательской группы в IBM, целью которой была проработка принципов интегрированных информационных систем в области законодательства. Плодом усилий этого коллектива, куда входили также Эдвард Мошер и Реймонд Лори, стал **GML** — обобщенный язык разметки (Generic Markup Language, Goldfarb-Mosher-Lorie). Многие решения этой группы нашли применение в различных издательских системах IBM. На основе GML и идей системы Scribe, разработанной Брайаном Рейдом, Международная организация по стандартизации (ISO) разработала метаязык **SGML** (стандарт ISO-8879:1986).

TeX

- Язык TeX (1979) Дональда Кнута на три года опередил Postscript, созданный Джоном Уорноком и др. в компании Adobe и предназначенный для точного описания внешней формы документов с композицией произвольной сложности. В отличие от низкоуровневого Postscript, язык TeX ориентировался на работу пользователей, не имеющих навыков программирования. Наиболее широко этот язык стал применяться в научной среде, где предъявляются самые высокие требования к качеству построения формул сложной структуры.

- Язык HTML, с появлением которого понятие «гипертекст» стало простым и обыденным,
- создавался на базе SGML путем максимального упрощения его структуры и свойств. Вряд ли
- имеет смысл описывать его подробно: это основной язык представления информации в Web-
- среде, включая Интернет.

XML и XHTML

- Метаязык XML, созданный во многом с подачи Джона Босака (Sun Microsystems), руководителя рабочей группы SGML ERB, также основывался на SGML, Это своего рода язык транспортирования и промежуточного хранения данных при обмене ими между разнородными и распределенными системами. На его основе можно проводить сколь угодно сложные преобразования документов и текстовой информации, а главное, в унифицированном виде хранить данные реляционно-иерархической структуры, в том числе по настройкам и программированию компонентов.
- В 2001 г. появилась ревизия HTML, которая получила название XHTML, где были учтены требования XML.

CFML и DHTML

- **CFML** (Cold Fusion) и **DHTML** — языки разметки, напрямую предназначенные для динамического создания Web-страниц и доступа к БД. Оба они являются расширениями HTML, но CFML — чисто коммерческий язык компании Allaire, не имеющий других реализаций, а DHTML, возникший из Dynamic HTML, — результат многочисленных компромиссов и конкуренции Netscape и Microsoft, которые были подытожены консорциумом W3C. Близкие им ASP (Active Server Pages, Microsoft) и JSP (Java Server Pages, Sun) — уже не языки, а, скорее, технологические «плавильные печи». Тот же ASP по сути не что иное, как сценарий на VBScript, который выполняется на сервере.

Универсальные сценарные языки

Название	Год появления	Разработчик языка	Организация, где был создан язык
Perl	1986	Ларри Уолл	UC Berkeley
Tcl	1990	Джон Устерхаут	
Python	1991	Гвидо ван Россум	Stichting Mathematisch Centrum
Ruby	1993	Юкиhiro Матсумото	Rapid Deployment Software
Euphoria	1993	Р. Крейг	
Lua	1994	У. Целес+	PUC-Rio
PHP	1995	Расмус Лердорф	Lucent Bell Labs
Mawl	1995	Д. Лэдд+	
Pike	1996	Фредерик Хьюбинетт	InformationsVavarna
Curl	2000	Стив Уард+	MIT Lab for Computer Science

Универсальные сценарные языки

- Представители этого класса языков наиболее широко известны. Именно их чаще всего и ассоциируют с термином «сценарный язык», причем применительно к Web-среде. При этом нередко упускается из виду тот факт, что создавались самые популярные из них совсем не для Web-программирования: языки Perl, Tcl и Python появились еще тогда, когда не было даже первой версии HTML.

«Сценарий, — это то, что вы передаете актеру, а программа — это то, что передается зрительному залу». Ларри Уолл,

Театр	Кино	Web-программирование
Режиссер	Режиссер-постановщик	Руководитель проекта
Автор пьесы	Автор сценария	Автор
Художник	Художник	Художник/дизайнер
Актер	Актер	-
-	Оператор	Дизайнер-технолог
-	Монтажер	Программист

можно провести простую аналогию между созданием Web-программ, театральным спектаклем и съемкой кинофильма.

- **Сценарий □ Спектакль.**

Сценарий в театре «интерпретируется» актерами. В итоге зрительный зал видит спектакль.

- **Сценарий □ Кинофильм.**

Сценарий в кино «компилируется» в съемочный материал, который монтажер компонует в фильм. В итоге зрительская аудитория видит кинофильм.

- **Сценарий □ Программа.**

Сценарий в Web-программировании автоматически (путем интерпретации/компиляции) превращается в программу. В итоге пользователь видит результат работы программы.

формы сценарных языков (их синтаксис)

- языки функционального программирования (Лисп, Scheme);
- языки обработки строк (Snobol, Icon);
- объектно-ориентированные языки (Smalltalk, Java, Eiffel, C#);
- языки управления заданиями, командные языки (csh, Rexx);
- языки управления средой (Tcl, VBA);
- языки разметки/макροобработки (SGML, TeX);
- языки моделирования дискретных систем (GPSS, SIMSCRIPT, SLAM II).

Какие требования предъявляются к сценарному языку?

- Он должен служить средством быстрого макетирования.
- Сценарный язык должен ориентироваться на скорость и простоту освоения базовых возможностей, быстро дающих видимый результат. Но из этого не следует, что язык должен быть примитивным.

Какие требования предъявляются к сценарному языку?

- Сценарный язык должен в меньшей степени опираться на создание конечного продукта с нуля и в большей степени — на использование тех мощностей, которыми обладает операционная система, графическая среда, прикладная сервисная машина и прочие подобные компоненты, вокруг которых строится «обвязка» в виде сценариев. Он в первую очередь обеспечивает удобную работу на уровне текстовых строк, стараясь по возможности не прибегать к интенсивному использованию сложных типов данных.