

Содержание:

ВВЕДЕНИЕ

Начало новой эпохи в программирование началось именно с языка С. Именно он определил направление всего будущего программирования.

Язык С++ - это расширение языка С. Его главное преимущество в том, что в С++ присутствует поддержка объектно-ориентированного программирования, перегруженных операций, а также наличие возможности разработать полномасштабные windows-приложения.

Современные технологии не стоят на месте. Тогда компания Sun Microsystems создает систему разработки Java, которая удовлетворяет всем требованиям.

Цель данной работы: ознакомиться с историей возникновения и развития языков программирования С и С++ и Java.

Задачи:

1. изучено программирование на С и С++: дано понятие языков программирования С и С++ их краткая история развития и базовые понятия;
2. изучено программирование Java: дано понятие языка программирования Java, их история и базовые понятия.

Глава 1 Программирование на С и С++

1.1 История возникновения и развития языка С++

Язык С создан в начале 70х годов, когда Кен Томпсон и Дэннис Ритчи из Bell Labs разрабатывали операционную систему UNDC. (рис. 1)



Рисунок 1. Создатели языка программирования C++

Сначала они создали часть компилятора C, затем использовали ее для компиляции остальной части компилятора C и, наконец, применили полученный в результате компилятор для компиляции UNIX. Операционная система UNIX первоначально распространялась в исходных кодах на C среди университетов и лабораторий, а получатель мог откомпилировать исходный код на C в машинный код с помощью подходящего компилятора C.

Распространение исходного кода сделало операционную систему UNIX уникальной. Программист мог изменить операционную систему, а исходный код мог быть перенесен с одной аппаратной платформы на другую.

По сравнению с более ранним языком — BCPL, C был улучшен путем добавления типов данных определенной длины. Например, тип данных `int` мог применяться для создания переменной с определенным числом битов (обычно 16), в то время как тип данных `long` мог использоваться для создания целой переменной с большим числом битов (обычно 32). В отличие от других языков высокого уровня, C мог работать с адресами памяти напрямую с помощью указателей и ссылок. Поскольку C сохранил способность прямого доступа к аппаратному обеспечению, его часто относят к языкам среднего уровня или в шутку называют «мобильным языком ассемблера».

Что касается грамматики и синтаксиса, то C является структурным языком программирования. В то время как многие современные программисты мыслят в категориях классов и объектов, программисты на C думают в категориях процедур

и функций. В С можно определить собственные абстрактные типы данных, используя ключевое слово `struct`. Аналогично можно описывать собственные целые типы (перечисления) и давать другие названия существующим типам данных при помощи ключевого слова `typedef`. В этом смысле С является структурным языком с зародышами объектно-ориентированного программирования.

Широкое распространение **языка С** на различных типах компьютеров (иногда называемых аппаратными платформами) привело, к сожалению, ко многим вариациям языка. Они были похожи, но несовместимы друг с другом. Это было серьезной проблемой для разработчиков программ, нуждавшихся в написании совместимых программ, которые можно было бы выполнять на нескольких платформах. Стало ясно, что необходима стандартная версия С. В 1983г. ANSI (Американский Национальный Комитет Стандартов) сформировал технический комитет X3J11 для создания стандарта языка С. В 1989 стандарт был утвержден. ANSI скооперировался с ISO (Международной Организацией Стандартов), чтобы стандартизировать С в международном масштабе. Этот стандарт совершенствуется до сих пор и поддерживается большинством фирм разработчиков компиляторов. В октябре 2011года появился новый стандарт языка С++ [Джесс либерти, «Освой самостоятельно С++ за 21 день», изд. Дом «Вильямс», Москва - Санкт-Петербург - Киев, 2001. - С.14.]

Бьерн Страуструп высвободил объектно-ориентированный потенциал С путем перенесения возможностей классов Simula 67 в С. Первоначально новый язык носил имя «С с классами» и только потом стал называться С++. Язык С++ достиг популярности, будучи разработанным в Bell Labs, позже он был перенесен в другие индустрии и корпорации. Сегодня это один из наиболее популярных языков программирования в мире. С++ наследует как хорошие, так и плохие стороны С.

Название **С++** выдумал Рик Масситти. Название указывает на эволюционную природу перехода к нему от С. «++» - это операция приращения в С. Чуть более короткое имя С+ является синтаксической ошибкой. Кроме того, оно уже было использовано как имя совсем другого языка. Знатоки семантики С находят, что С++ хуже, чем ++С. Названия D язык не получил, поскольку он является расширением С и в нем не делается попыток исцеляться от проблем путем выбрасывания различных особенностей.

В **языке С++** полностью поддерживаются принципы объектно-ориентированного программирования, включая три кита, на которых оно стоит: инкапсуляцию, наследование и полиморфизм. Инкапсуляция в С++ поддерживается посредством

создания нестандартных (пользовательских) типов данных, называемых классами.

Язык C++ поддерживает наследование. Это значит, что можно объявить новый тип данных (класс), который является расширением существующего языка.

Язык программирования C++ в настоящее время считается господствующим языком, используемым для разработки коммерческих продуктов, 90% игр пишутся на C++ с применением DirectX.

1.2 Достоинства и недостатки языка

В каждом языке есть свои плюсы и минусы. Для работы с языком C++ их необходимо учитывать, тогда трудностей в написании программ не возникнет. Зная все тонкости языка, специалист в этой области всегда будет востребован.

Достоинства языка:

1. Масштабируемость. На языке C++ разрабатывают программы для самых различных платформ и систем.
2. Возможность работы на низком уровне с памятью, адресами, портами.
3. Возможность создания обобщенных алгоритмов для разных типов данных, их специализация, и вычисления на этапе компиляции, используя шаблоны.
4. Кроссплатформенность. Доступны компиляторы для большого количества платформ, на языке C++ разрабатывают программы для самых различных платформ и систем.
5. Эффективность. Язык спроектирован так, чтобы дать программисту максимальный контроль над всеми аспектами структуры и порядка исполнения программы.

Недостатки языка:

1. Наличие множества возможностей, нарушающих принципы типобезопасности приводит к тому, что в C++ программы могут легко закрасться трудноуловимая ошибка.
2. Плохая поддержка модульности. Подключение интерфейса внешнего модуля через препроцессорную вставку заголовочного файла (`#include`) серьезно

замедляет компиляцию, при подключении большого количества модулей.

3. Язык C++ является сложным для изучения и для компиляции.

4. Некоторые преобразования типов неинтуитивны. В частности, операция над беззнаковым и знаковым числами выдаёт беззнаковый результат.

5. Некоторые считают недостатком языка C++ отсутствие встроенной системы сборки мусора. С другой стороны, в C++ имеется достаточно средств, позволяющих почти исключить использование опасных указателей, нет принципиальных проблем и в реализации и использовании сборки мусора (на уровне библиотек, а не языка). Отсутствие встроенной сборки мусора позволяет пользователю самому выбрать стратегию управления ресурсами.

1.3 Базовые понятия языка

Алфавитом языка называется совокупность символов, используемых в языке.

Очень важно знать и помнить, что язык C различает прописные и строчные буквы. Язык C, как говорят, является чувствительным к регистру. В языке C имена PEREMANNAY, Peremennay и peremennay имеют три различных имени переменных. При написании программы необходимо внимательно относиться к использованию регистров для имен переменных. Удобнее всего принять некоторые соглашения относительно использования прописных и строчных букв в идентификаторах. Например, имена переменных содержат только строчные буквы, константы и макросы – прописные буквы.

В языке C++ алфавит можно условно разделить на группы знаков. [Джесс либерти, «Освой самостоятельно C++ за 21 день», изд. Дом «Вильямс», Москва - Санкт-Петербург - Киев, 2001.- С.43.]:

- прописные и строчные латинские буквы (A, B, ..., Z, a, b, ..., z);
- арабские цифры (0, 1, 2, 3, 4, 5, 6, 7, 8, 9);
- специальные знаки: " , { } | [] () + - / % \ ; ' . : ? < = > _ ! & * # ~ ^
- пробельные символы – это символы пробела, табуляции, перехода на новую строку.

Идентификатор – это имя какого-либо объекта программы. Имя может состоять из латинских букв, цифр и символа _ (символ подчеркивания). Имя может начинаться с буквы или символа подчеркивания, но не с цифры.

Длина имени в большинстве случаев не ограничена, но на длину внешних имён (имён библиотек, функций) может накладываться ограничение операционная система.

Ключевые (зарезервированные) слова – это слова, которые можно использовать в программе только по их прямому назначению, т.е. эти имена нельзя использовать для обозначения, например, своих переменных [Герберт Шилдт. С++ для начинающих. Пер. с англ. М: ЭКОМПаблишерз 2007. - С.34-38.].

Ключевые слова языка C++

asm	auto	bool	break
case	catch	char	class
const	const_cast	continue	default
delete	do	double	dynamic_cast
else	enum	explicit	export
extern	false	float	for
friend	goto	if	inline
int	long	mutable	namespace
new	operator	private	protected
public	register	reinterpret_cast	return
short	signed	sizeof	static
static_cast	struct	switch	template
this	throw	true	try
typedef	typeid	typename	union

В языке C++ имеется 63 ключевых слова. (Рис.2)

Рисунок 2. Ключевые слова языка C++

Константы – это неизменяемые величины. Типы констант в языке C++: целые, вещественные, символьные, строковые и булевы [Подбельский В.В., Фомин С.С. Программирование на языке Си. Учеб. пособие.- М.: Финансы и статистика, 2000.]:

1. Целые константы. Правила языка допускают использование трёх видов целых констант: десятичных, шестнадцатеричных и восьмеричных. Основание определяется префиксом в записи константы. Для десятичных констант префикс не нужен.

Десятичное целое – это последовательность десятичных цифр, начинающаяся не с нуля (если это не число ноль).

Восьмеричное целое – это последовательность цифр, начинающаяся с 0 и не содержащая десятичных цифр старше 7.

Шестнадцатеричное целое – это последовательность шестнадцатеричных цифр (0, 1, ..., 9, A, B, C, D, E, F), перед которой записаны символы 0x или 0X.

2. Вещественная константа. Константа с плавающей точкой (вещественная константа) всегда представляется числом с плавающей точкой двойной точности, т. е. как имеющая тип `double`, и состоит из следующих частей:

- целой части — последовательности цифр;
- точки — разделителя целой и дробной части;
- дробной части — последовательности цифр;
- символа экспоненты `e` или `E`;
- экспоненты в виде целой константы (может быть со знаком).

Любая часть (но не обе сразу) из нижеследующих пар может быть опущена:

- целая или дробная часть;
- точка или символ `e` (`E`) и экспонента в виде целой константы.

По умолчанию компилятор присваивает вещественному числу тип `double`. Если программиста не устраивает тип, который компилятор приписывает константе, то тип можно явно указать в записи константы с помощью следующих суффиксов:

- `F` (или `f`) — `float` для простых вещественных констант,
- `L` (или `l`) — `long double` для вещественных констант двойной расширенной точности.

3. Символьные константы

Символьная константа — это один символ, например: `'z'`. В качестве символьных констант также могут использоваться управляющие коды, не имеющие

графического представления. При этом код управляющего символа начинается с символа ‘\’ (обратный слеш).

Таблица 1.

Управляющие символы

Код	Обозначение	Описание
0x00	‘\0’	Нуль-символ, NULL
0x07	‘\a’	Звуковой сигнал.
0x08	‘\b’	Возврат на 1 шаг (Backspace)
0x09	‘\t’	Горизонтальная табуляция (Tab)
0x0A	‘\n’	Перевод строки (Enter)
0x0B	‘\v’	Вертикальная табуляция (в консоли аналогична переводу строки)
0x0C	‘\f’	Смена страницы
0x0D	‘\r’	Возврат каретки

При нажатии клавиши **Enter** генерирует сразу два управляющих символа – перевод строки (0x0A) и возврат каретки (0x0D).

Все символьные константы имеют тип char и занимают в памяти 1 байт. Значением символьной константы является числовое значение её внутреннего кода.

4.Строковые константы

Строковая константа — это последовательность символов, заключенная в кавычки, например:

«Строковая константа»

Кавычки не входят в строку, а лишь ограничивают её. Технически строковая константа представляет собой массив символов, и по этому признаку может быть отнесена к разряду сложных объектов языка Си.

В конце каждой строковой константы компилятор помещает '\0' (нуль-символ), чтобы программе было возможно определить конец строки. Такое представление означает, что размер строковой константы не ограничен каким-либо пределом, но для определения длины строковой константы её нужно полностью просмотреть.

Поскольку строковая константа состоит из символов, то она имеет тип `char`. Количество ячеек памяти, необходимое для хранения строковой константы на 1 больше количества символов в ней (1 байт используется для хранения нуль-символа).

Символьная константа 'x' и строка из одного символа «x» — не одно и то же. Символьная константа — это символ, используемый для числового представления буквы x, а строковая константа «x» содержит символ 'x' и нуль-символ '\0' и занимает в памяти 2 байта. Если в программе строковые константы записаны одна за другой через разделители, то при выполнении программы они будут размещаться в последовательных ячейках памяти.

5. Булевы константы. Имеется всего два значения: `true` -- истина и `false` – ложь.

Комментарий – это последовательность любых знаков (символов), которая используется в тексте программы для её пояснения. Обычно в тексте программы делают вводный комментарий к программе в целом (её назначение, автор, дата создание и т.д.), а далее дают комментарии к отдельным фрагментам текста программы, смысл которых не является очевидным. Важно дать объяснение не только того, что делается, а с какой целью это делается. Комментарии важны всегда: будь то создание программа для себя, или с ней будут работать другие.

Комментарии компилятор не считывает, они имеют значение только для пользователя. В языке C++ имеется два вида комментариев: однострочные и многострочные.

Однострочный комментарий начинается с символов // (две косые черты). Всё, что записано после этих символов и до конца строки, считается комментарием.

Например:

```
// Текст комментария
```

Многострочный комментарий начинается парой символов /* (косая черта и звёздочка) и заканчивается символами */ (звёздочка и косая черта). Текст такого комментария может занимать одну или несколько строк. Всё, что находится между знаками /* и */, считается комментарием. Например:

```
/* Здесь находится
```

```
многострочный комментарий */
```

Вывод: В результате анализа литературных источников в первой главе были выделены следующие разделы:

- 1. История возникновения и развитие языка C и C++: В этом разделе мы рассмотрели понятие языка C и C++. Изучили историю создания и выделили этапы развития языка. Язык C++ является одним из ведущих языков программирования и находится на первых местах в рейтинге.
- 2. Достоинства и недостатки языка: Как и у любого другого языка есть достоинства и недостатки, но у языка C и C++ достоинства видны сразу при работе. А именно: масштабируемость, возможность работы на низком уровне с памятью, адресами, портами, возможность создания обобщенных алгоритмов для разных типов данных, кроссплатформенность, эффективность.
- 3. Основные понятия: В данном разделе познакомились с алфавитом языка, типами констант и ключевыми словами.

Глава 2. Программирование на Java

2.1 История создания и развития языка Java

Java - язык программирования, который был разработан в компании Sun Microsystems. Приложения Java обычно компилируются в специальный байт-код, вследствие чего они имеют возможность работать на любой виртуальной Java-

машине (JVM) независимо от самой компьютерной архитектуры. Дата официального выпуска языка - 23 мая 1995 года. Сегодня технология Java предоставляет средства для превращения статических Web-страниц в интерактивные динамические документы, а также возможности создания распределенных (не зависящих от платформы) приложений.

Рождению языка Java предшествовала довольно интересная история. В 1990 году разработчик ПО компании Sun Microsystems Патрик Нотон (Patrick Naughton) понял, что ему надоело поддерживать сотни различных интерфейсов программ, используемых в компании, и сообщил исполнительному директору Sun Microsystems и своему другу Скотту МакНили (ScottMcNealy) о своем намерении перейти работать в компанию NeXT. МакНили, в свою очередь, попросил Нотона составить список причин своего недовольства и выдвинуть такое решение проблем, как если бы он был Богом и мог исполнить все, что угодно.

Нотон, хотя и не рассчитывал на то, что кто-то обратит внимание на его письмо, все же изложил свои претензии, беспощадно раскритиковав недостатки SunMicrosystems, в частности, разрабатываемую в тот момент архитектуру ПО NeWS. К удивлению Нотона, его письмо возымело успех: оно было разослано всем ведущим инженерам SunMicrosystems, которые не замедлили откликнуться и высказать горячую поддержку своему коллеге и одобрение его взглядов на ситуацию в SunMicrosystems. Обращение вызвало одобрение и у высшего руководства компании, а именно, у Билла Джоя (BillJoy), основателя SunMicrosystems, и Джеймса Гослинга (JamesGosling), начальника Нотона.

В тот день, когда Нотон должен был уйти из компании, было принято решение о создании команды ведущих разработчиков с тем, чтобы они делали что угодно, но создали нечто необыкновенное.

Команда из шести человек, с кодовым названием Green, ушла в самовольное изгнание, погрузившись в исследования бытовых устройств, таких как NintendoGameBoys, устройств дистанционного управления. Команда Green пыталась найти средство, с помощью которого можно было бы установить взаимодействие между этими устройствами. Вскоре стало ясно, что такие электроприборы, как видеомэгафоны, проигрыватели лазерных дисков, стереосистемы - все они были реализованы на разных процессорах. Это означало, что если производитель захочет добавить телевизору или видеомэгафону дополнительные функции или характеристики, он будет зажат в рамках средств, зашитых в аппаратное обеспечение. Эта проблема, в сочетании с ограниченностью

памяти микросхем этих устройств, выдвинула новый подход к программированию ПО, который должен был стать ведущим на рынке бытовой электроники.

Команда приступила к разработке нового объектно-ориентированного языка программирования, который был назван Oak (дуб), в честь дерева, росшего под окном Гослинга.

Вскоре компания SunMicrosystems преобразовала команду Green в компанию FirstPerson. Новая компания обладала интереснейшей концепцией, но не могла найти ей подходящего применения. После ряда неудач неожиданно ситуация для компании резко изменилась: был анонсирован Mosaic - так родился WorldWide Web, с которого началось бурное развитие Internet.

Нотон предложил использовать Oak в создании Internet- приложений. Так Oak стал самостоятельным продуктом, вскоре был написан Oak-компилятор и Oak-браузер "WebRunner". В 1995 году компания SunMicrosystems приняла решение объявить о новом продукте, переименовав его в Java (единственное разумное объяснение названию - любовь программистов к кофе). Когда Java оказалась в руках Internet, стало необходимым запускать Java-апплеты - небольшие программы, загружаемые через Internet. WebRunner был переименован в HotJava и компания Netscape встала на поддержку Java-продуктов. [Джон Родли Создание Java-апплетов .- TheCoriolis Group,Inc.,1996, Издательство НИПФ "ДиаСофт Лтд.",1996 - С.32].

Java - это название не только самого языка, но и платформы для создания и исполнения приложений на основе данного языка.

Изначально язык разрабатывался для программирования бытовых электронных устройств. Затем он стал применяться для написания клиентских приложений и серверного ПО. Название было дано в честь марки кофе Java, который так любили некоторые программисты, поэтому на официальной эмблеме языка и изображена чашка с дымящимся кофе [Джон Родли Создание Java-апплетов .- TheCoriolis Group,Inc.,1996, Издательство НИПФ "ДиаСофт Лтд.",1996 - С.35-37.].

«За полтора года мы сделали столько же, сколько иные большие коллективы в Sun делали за три года, - с гордостью отмечает инженер Нотон. - Операционную систему GreenOS, язык, инструментарий, пользовательский интерфейс, новую аппаратную платформу, три заказных микросхемы и каждый этап был связан с риском, т.к. мы использовали совершенно новые технологии».

В 1992 г. в рамках проекта «Green» была выпущена первая продукция под названием «*7». Это было устройство интеллектуального дистанционного управления. (Умещааясь в корпусе 6x4x4 дюйма, оно имело мощность рабочей станции SPARC.)

До появления Java страница WWW по факту представляла собой листок бумаги. С появлением Java браузер задает структуру и резко расширяет возможности провайдеров содержания. «В границах этой модели трудно создавать распределенные системы и осуществлять их модернизацию, в частности, если их элементы имеют разное происхождение» - указывает Гослинг.

Достоинства и недостатки языка

Программы на Java транслируются в байт-код, которые выполняет виртуальная машина Java (JVM) - программа, которая обрабатывает байтовый код и передает далее инструкции оборудованию в роли интерпретатора. Несмотря на то, что появились другие языки, **Java не отстает от времени**. [Майкл Эферган Java: справочник.- QUE Corporation, 1997, Издательство «Питер Ком», 1998. - 687с.]. Благодаря ООП (объектно-ориентированному программированию), где можно определить данные по типу и структуре, разобраться с применяемыми функциями и управлять ими. ООП позволяет:

- Повторное применение объектов в различных программах.
- Предотвращать ошибки - объект скрывает информацию, запрещенную к свободному доступу.
- Организовать структуру программы.
- Упростить обслуживание при модернизации старых кодов.

Java принадлежит к механизму, в котором достигнут **простой синтаксис** и **плавная кривая обучения**. Его преобразование происходит компиляторами и интерпретаторами, отчего упрощенная разработка сделала его легким, чтобы писать, читать и обслуживать.

Достоинства языка является:

- Использование в **корпоративных приложениях**, Java способен поддерживать строительные блоки системы или различные библиотеки, с их помощью создают необходимые функции.

- **Запуск приложений в «песочнице»** с устранением распространённых, уязвимых объектов в соответствии с политикой безопасности.
- Независимое положение от платформ.
- **Повышенная производительность** труда благодаря встроенному механизму, чтобы совместно использовать данные программы на нескольких компьютерах.
- Возможность **автоматического управления памятью** с одновременным независимым запуском потоков.

Разработчики не остановились в своих изысканиях, по-прежнему идет выпуск новых версий, функции которых остаются интересны программистам.

Недостатки языка:

- Простым пользователям версии Java изначально предоставлялись бесплатно. Но в 2019 году компанией Oracle объявлено, что теперь они начнут **брать плату за коммерческое использование языка**. Начнется оценка, кто и с какой целью пользуется Java. Потребители в свою очередь будут вести поиск альтернативных решений.
- **Низкая скорость и безопасность**. Все языки с высоким уровнем страдают малой производительностью, этому способствуют различные функции – очистка памяти, настройки, блокировки.
- **Многословность и сложность кода**. Язык с длинными, трудными предложениями помогает при его изучении. Но лишняя информация затрудняет чтение. Поэтому в среде программистов Java считается слишком громоздким.
- **Популярность не всегда играет хорошую роль**. При повсеместном использовании возникает угроза в появлении более удобных объектов, которые решают те же задачи, но с большей скоростью и меньшим количеством ошибок. Тогда, созданные программы станут бесполезными, а на них затрачены ресурсы, время.
- Одним из основных минусов языка является его **большой объем потребления памяти** даже, когда не проводятся сложные операции. Но в оправдание инженеров Java можно выставить постоянные работы по оптимизации, где разработчики добились успехов в сокращении проблем с памятью.

Основные понятия языка

Идеи, которые лежат в основе концепции и различных реализациях среды виртуальной машины Java, дали вдохновение для множества энтузиастов на расширение перечня языков, которые можно было бы использовать для создания программ, которые исполняли на виртуальной машине. Эти идеи нашли также выражение в спецификации общеязыковой инфраструктуры CLI, заложенной в основу платформы.NET компанией Microsoft. [Васильев А. Н. «Программирование на Java для начинающих», изд. Эскиммо, 2017. – 704с.]

Основные возможности:

- автоматическое управление памятью;
- расширенные возможности обработки исключительных ситуаций;
- богатый набор средств фильтрации ввода/вывода;
- набор стандартных коллекций, таких как массив, список, стек и т. п.;
- наличие простых средств создания сетевых приложений (в том числе с использованием протокола RMI);
- наличие классов, позволяющих выполнять HTTP-запросы и обрабатывать ответы;
- встроенные в язык средства создания многопоточных приложений;
- унифицированный доступ к базам данных:
 - на уровне отдельных SQL-запросов - на основе JDBC, SQLJ;
 - на уровне концепции объектов, обладающих способностью к хранению в базе данных - на основе Java Data Objects и Java Persistence API;
- поддержка шаблонов (начиная с версии 1.5);
- параллельное выполнение программ.

2.3 Базовые понятия языка

Алфавит языка Java состоит из букв, десятичных цифр и специальных символов. Буквами считаются латинские буквы (кодируются в стандарте ASCII), буквы национальных алфавитов (кодируются в стандарте Unicode, кодировка UTF-16), а также соответствующие им символы, кодируемые управляющими последовательностями

Буквы и цифры можно использовать в качестве идентификаторов (т.е. имен) переменных, методов и других элементов языка программирования. Правда, при использовании в идентификаторах национальных алфавитов в ряде случаев могут возникнуть проблемы - эти символы будут показываться в виде вопросительных знаков.

Как буквы рассматривается только часть символов национальных алфавитов. Остальные символы национальных алфавитов - это специальные символы. Они используются в качестве операторов и разделителей языка Java и не могут входить в состав идентификаторов.

Латинские буквы ASCII

- ABCD...XYZ - заглавные (прописные) ,
- abcd...xyz - строчные

Дополнительные "буквы" ASCII

- _ - знак подчеркивания,
- \$ - знак доллара.

Национальные буквы на примере русского алфавита

- АБВГ...ЭЮЯ - заглавные (прописные),
- абвг...эюя - строчные
- Десятичные цифры 0 1 2 3 4 5 6 7 8 9

Десятичные и шестнадцатеричные цифры и целые числа

Целые числовые константы [Майкл Эферган]Java: справочник .- QUE Corporation, 1997, Издательство "Питер Ком", 1998. - С.89.] в исходном коде Java (так называемые литерные константы) могут быть десятичными или шестнадцатеричными. Они записываются либо символами ASCII, или символами Unicode следующим образом.

Десятичные константы записываются как обычно. Например, -347.

Шестнадцатеричная константа начинается с символов 0x или 0X (цифра 0, после которой следует латинская буква X), а затем идет само число в шестнадцатеричной нотации.

Целая константа в обычной записи имеет тип `int`. Если после константы добавить букву `L` (или `l`, что хуже видно в тексте, хотя в среде разработки выделяется цветом), она будет иметь тип `long`, обладающий более широким диапазоном значений, чем тип `int`.

Поясним теперь, что такое шестнадцатеричная нотация записи чисел и зачем она нужна.

Информация представляется в компьютере в двоичном виде - как последовательность бит. Бит - это минимальная порция информации, он может быть представлен в виде ячейки, в которой хранится или ноль, или единица. Но бит - слишком мелкая единица, поэтому в компьютерах информация хранится, кодируется и передается байтами - порциями по 8 бит.

Под «ячейкой памяти» будет пониматься непрерывная область памяти (с последовательно идущими адресами), выделенная программой для хранения данных. На рисунках мы будем изображать ячейку прямоугольником, внутри которого находятся хранящиеся в ячейке данные. Если у ячейки имеется имя, оно будет писаться рядом с этим прямоугольником.

Мы привыкли работать с числами, записанными в так называемой десятичной системе счисления. В ней имеется 10 цифр (от 0 до 9), а в числе имеются десятичные разряды. Каждый разряд слева имеет вес 10 по сравнению с предыдущим, то есть для получения значения числа, соответствующего цифре в каком-то разряде, стоящую в нем цифру надо умножить на 10 в соответствующей степени.

В программировании десятичной системой счисления пользоваться не всегда удобно, так как в компьютерах информация организована в виде бит, байт и более крупных порций [Шильдт Герберт «Java. Руководство для начинающих. Современные методы создания, компиляции и выполнения программ на Java» , 2018.- 816с.]. Человеку неудобно оперировать данными в виде длинных последовательностей нулей и единиц. В настоящее время в программировании стандартной является шестнадцатеричная система записи чисел. Например, с ее

помощью естественным образом кодируется цвет, устанавливаются значения отдельных бит числа, осуществляется шифрование и дешифрование информации, и так далее. В этой системе счисления все очень похоже на десятичную, но только не 10, а 16 цифр, и вес разряда не 10, а 16. В качестве первых 10 цифр используются обычные десятичные цифры, а в качестве недостающих цифр, больших 9, используются заглавные латинские буквы A, B, C, D, E, F:

0 1 2 3 4 5 6 7 8 9 A B C D E F

То есть A=10, B=11, C=12, D=13, E=14, F=15.

Заметим, что в шестнадцатеричной системе счисления числа от 0 до 9 записываются одинаково, а превышающие 9 отличаются. Для чисел от 10 до 15 в шестнадцатеричной системе счисления используются буквы от A до F, после чего происходит использование следующего шестнадцатеричного разряда.

Зарезервированные слова языка Java [Майкл Эферган Java: справочник .- QUE Corporation, 1997, Издательство "Питер Ком", 1998 - С.67].

Это слова, зарезервированные для синтаксических конструкций языка, причем их назначение нельзя переопределять внутри программы (см. табл.2)

Таблица 2

Слова, зарезервированные для синтаксических конструкций языка

abstract	boolean	break	Byte	Case
Catch	char	class	Const	continue
default	do	double	Else	Enum
extends	false	final	Finally	Float
For	goto	if	Implements	import

instanceof	int	interface	Long	Native
New	null	package	Private	protected
public	return	short	Static	Super
switch	synchronized	this	Throw	throws
transient	true	try	Void	volatile

While

Их нельзя использовать в качестве идентификаторов (имен переменных, подпрограмм и т.п.), но можно использовать в строковых выражениях.

В языке Java только 8 примитивных (скалярных, простых) типов: boolean, byte, char, short, int, long, float, double [Эфеган Java: справочник .- QUE Corporation, 1997, Издательство «Питер Ком», 1998. - С.71-75.]

В языке Java действуют следующие правила:

1. Если один операнд имеет тип double, другой тоже преобразуется к типу double.
2. Иначе, если один операнд имеет тип float, другой тоже преобразуется к типу float.
3. Иначе, если один операнд имеет тип long, другой тоже преобразуется к типу long.
4. Иначе оба операнда преобразуются к типу int.

Данный способ неявного преобразования встроенных типов полностью совпадает с преобразованием типов в C++.

Вывод: В результате анализа литературных источников во второй главе были выделены следующие разделы:

История возникновения и развитие языка Java: В этом разделе мы рассмотрели понятие языка Java. Изучили историю создания и выделили этапы развития языка.

Достоинства языка:

- Использование в **корпоративных приложениях**, Java способен поддерживать строительные блоки системы или различные библиотеки, с их помощью создают необходимые функции.
- **Запуск приложений в «песочнице»** с устранением распространённых, уязвимых объектов в соответствии с политикой безопасности.
- Независимое положение от платформ.
- **Повышенная производительность** труда благодаря встроенному механизму, чтобы совместно использовать данные программы на нескольких компьютерах.
- Возможность **автоматического управления памятью** с одновременным независимым запуском потоков.

Основные понятия: В данном разделе познакомились с алфавитом языка, типами констант и ключевыми словами.

ЗАКЛЮЧЕНИЕ

Благодаря языку C++ произошел стремительный прорыв в развитии всего программирования. C++ и сегодня занимает доминирующее положение среди всех языков программирования в мире.

Благодаря ему множество программистов разрабатывает огромное число различных проектов. И в будущем этот язык программирования сохранит свои позиции, совершенствуясь изо дня в день

Язык Java является объектно-ориентированным и поставляется с достаточно объемной библиотекой классов. Благодаря библиотекам классов Java значительно упростилась разработка приложений, ведь в распоряжение программиста предоставлены мощные средства решения распространенных задач. Вследствие этого программист имеет возможность больше внимания уделить решению прикладных задач, а не таких, как, например, организация динамических массивов, взаимодействие с ОС или реализация элементов пользовательского интерфейса.

Свойства языка Java:

- язык программирования объектно-ориентирован, оснащён богатой библиотекой классов и одновременно довольно прост для освоения;
- цикл разработки приложений сокращен при помощи того, что система построена на основе интерпретатора;
- приложение получается автоматически переносимым между множеством платформ и ОС;
- за счет встроенной системы сборки «мусора» программист освобождается от необходимости явного управления памятью;
- приложение легко сопровождается и модифицируется, т.к. модули могут быть загружены с сети;
- в приложения встроена система безопасности, не допускающая незаконного доступа и проникновения вирусов.

Библиографический список

1. Х.М.Дейтел, П.Дж.Дейтел «Как программировать на C++», изд. Бином, М. – 2017. - 100с.
2. Симон Робинсон, Олли Корнес, Джей Глинн и др. «С# для профессионалов», изд. Лори – 2003г.
3. Бьерн Страуструп. «Язык программирования C++» , изд. Бином, М. -1136с.
4. Джесс Либерти, «Освой самостоятельно C++ за 21 день», изд. Дом «Вильямс», Москва - Санкт-Петербург - Киев, 2001. - 834с.
5. Джон Родли Создание Java-апплетов.- TheCoriolis Group,Inc.,1996, Издательство НИПФ «ДиаСофт Лтд.»,1996 - 654с.
6. Герберт Шилдт. C++ для начинающих. Пер. с англ. М: ЭКОМПублишерз 2007. - 546с.
7. Герберт Шилдт. Полный справочник по C + + 4-е издание. Пер. с англ. М: издательский дом « Вильямс » 2010. - 543с.
8. Майкл Эферган Java: справочник.- QUE Corporation, 1997, Издательство «Питер Ком», 1998. - 687с.
9. Васильев А. Н. «Программирование на Java для начинающих», изд. Эскимо, 2017. – 704с.

10. Шильдт Герберт «Java. Руководство для начинающих. Современные методы создания, компиляции и выполнения программ на Java» , 2018.- 816с.
11. <https://infourok.ru/statya-na-temu-ava-kak-moschneyshiy-yazik-programmirovaniya-dlya-razrabotki-interaktivnih-produktov-dlya-seti-internet-3145341.html>
12. <https://infourok.ru/nauchnaya-statya-na-temu-izucheniya-yazika-obektnoorientirovannogo-programmirovaniya-3165925.html>