

Совершит ли революцию язык Java в области клиент-серверных приложений?

Содержание.

1. Тезисы.....	2
2. Язык Java. История создания.....	3
3. Распространенные заблуждения относительно языка Java.....	6
4. Программная среда. Установка на компьютер.....	9
5. Обзор IDE средств для программирования на Java.....	13
6. Типы данных	17
7. Самые популярные ЯП в России.....	18
8. Игра, созданная на Java.....	19
9. Список используемой литературы.....	22

Цели работы:

1. Знать об истории создания Java.
2. Изучить возможности языка программирования.
3. Разработать собственную игру на языке Java.

Задачи работы:

1. Изучить программную среду.
2. Знать типы данных в Java.
3. Напомнить о том, что программирование на языке Java-перспективное занятие.

Объект исследования: Язык программирования Java.

Практическая значимость: работы определяется возможностью использовать представленные материалы на занятиях. Текст данного доклада может служить для расширения кругозора студентов.

Язык Java. История создания.

История Java восходит к 1991 году, когда группа инженеров из компании Sun под руководством Патрика Нотона (Patrick Naughton) и члена Совета директоров (и разностороннего компьютерного волшебника) Джеймса Гослинга (James Gosling) занялась разработкой небольшого языка, который можно было бы использовать для программирования бытовых устройств, например, контроллеров для переключения каналов кабельного телевидения (cable TV switchboxes).

Поскольку разные производители могут выбирать разные центральные процессоры (Central Processor Unit— CPU), было важно не завязнуть в какой-то одной архитектуре компьютеров. Проект получил кодовое название «Green».

Стремясь изобрести небольшой, компактный и машиннезависимый код, разработчики возродили модель, использованную при реализации первых версий языка Pascal на заре эры персональных компьютеров. Никлаус Вирт, создатель языка Pascal, в свое время разработал машиннезависимый язык, генерирующий промежуточный код для некоей гипотетической машины. Этот язык стал коммерческим продуктом под названием UCSD Pascal. (Такие гипотетические машины часто называются виртуальными — например, виртуальная машина языка Java, или JVM.) Этот промежуточный код можно выполнять на любой машине, имеющей соответствующий интерпретатор. Инженеры, работавшие над проектом «Green», также использовали виртуальную машину, что решило их основную проблему.

Однако большинство сотрудников компании Sun имели опыт работы с операционной системой UNIX, поэтому в основу разрабатываемого ими языка был положен язык C++, а не Pascal. В частности, они сделали язык объектно-, а не процедурно-ориентированным.

Как сказал Гослинг в своем интервью: *«Язык — это всегда средство, а не цель»*. Сначала Гослинг решил назвать его «Oak» («Дуб»). (Возможно потому, что он любил смотреть на дуб, растущий прямо под окнами его офиса в компании Sun.) Потом сотрудники компании Sun узнали, что слово Oak уже используется в качестве имени ранее созданного языка программирования, и изменили название Java.

В 1992 году в рамках проекта Green была выпущена первая продукция, названная «*7». Это было средство для чрезвычайно интеллектуального дистанционного управления. (Оно имело

мощность рабочей станции SPARK, помещаясь в коробочке размером 6x4x4 дюйма.) К сожалению, ни одна из компаний— производителей электронной техники не заинтересовалась этим изобретением.

Затем группа стала заниматься разработкой устройства для кабельного телевидения, которое могло бы осуществлять новые виды услуг, например, включать видеосистему по требованию. И снова они не получили ни одного контракта. (Забавно, что одной из компаний, отказавшихся подписать с ними контракт, руководил Джим Кларк (Jim Clark) — основатель компании Netscape, впоследствии сделавшей очень много для успеха языка Java.

Весь 1993 год и половину 1994 года продолжались безрезультатные поиски покупателей продукции, разработанной в рамках проекта «Green» (под новым названием «First Person, Inc.»). (Патрик Нотон, один из основателей группы, впоследствии в основном занимавшийся маркетингом, налетал в общей сложности более 300 тысяч миль, пытаясь продать разработанную технологию.) Проект «First Person, Inc.» Был прекращен в 1994 году.

Тем временем в рамках Интернет разрасталась сеть World Wide Web. Ключом к этой сети является браузер, превращающий гипертекст в изображение на экране.

В 1994 году большинство людей пользовалось браузером Mosaic, некоммерческим Web-браузером, разработанным в суперкомпьютерном центре Университета штата Иллинойс (University of Illinois) в 1993 году. (Частично этот браузер был написан Марком Андреессеном (Mark Andreessen) за 6,85 доллара в час. В то время Марк заканчивал университет и браузер был его дипломной работой. Затем он стал одним из основателей и главным программистом компании Netscape, и к нему пришли слава и богатство.)

В своем интервью журналу Sun World Гослинг сказал, что в середине 1994 года разработчики языка поняли: «Нам нужно создать действительно крутой браузер. Такой браузер должен представлять собой одно из немногих приложений модной клиент-серверной технологии, в которой жизненно важным было бы именно то, что мы сделали: архитектурная независимость, выполнение в реальном времени, надежность, безопасность — вопросы, не являвшиеся чрезвычайно важными для рабочих станций. И мы создали такой браузер».

На самом деле браузер был разработан Патриком Нотоном и Джонатаном Пэйном (Johnatan Payne). Позднее он превратился в современный браузер HotJava. Этот браузер был написан на языке Java, чтобы продемонстрировать всю его мощь. Однако разработчики не забывали о мощных средствах, которые теперь называются апплетами, наделив свой браузер способностью выполнять код внутри Web-страниц. «Демонстрация технологии» была представлена на выставке Sun World '95 23 мая 1995 года и вызвала всеобщее помешательство на почве языка Java, продолжающееся и поныне.

Компания Sun выпустила первую версию языка Java в начале 1996 года. Через несколько месяцев после нее появилась версия Java 1.02. Люди быстро поняли, что версия Java 1.02 не подходит для разработки серьезных приложений. Конечно, эту версию можно применять для разработки Web-страниц с пляшущими человечками, однако в версии Java 1.02 ничего нельзя даже напечатать. Честно говоря, версия Java 1.02 была еще сырой. Ее преемница, версия Java 1.1, заполнила большинство зияющих провалов, намного улучшив возможность отражения и добавив новую модель событий для программирования графического пользовательского интерфейса. Несмотря на это, она все еще была довольно ограниченной.

Выпуск версии Java 1.2 стал основной новостью конференции JavaOne в 1998 году. В новой версии слабые средства для создания графического пользовательского интерфейса и графических приложений были заменены сложным и масштабным инструментарием. Это был шаг вперед, к реализации лозунга «Write Once, Run Anywhere»™ («Один раз напиши— и везде выполняй»), выдвинутого при разработке предыдущих версий. В декабре 1998 года через три дня после выхода в свет название новой версии было изменено на громоздкое словосочетание Java 2 Standart Edition Software Development Kit Version 1.2 (Стандартное издание пакета инструментальных средств для разработки программного обеспечения на языке Java 2, версия 1.2).

Кроме стандартного издания пакета («Standart Edition») были предложены еще два варианта: «микроиздание» («Micro Edition») для портативных устройств, например, для мобильных телефонов, и «промышленное издание» («Enterprise Edition») для создания серверных приложений. В нашей книге в центре внимания находится стандартное издание. Версии 1.3 и 1.4 стандартного издания пакета инструментальных средств намного совершеннее первоначального выпуска языка Java 2. Они обладают новыми возможностями и, разумеется, содержат намного меньше ошибок. В табл. 1.1 показан стремительный рост объема библиотеки API по мере появления новых версий стандартного издания пакета SDK.

Таблица 1.1. Рост объема библиотеки API из пакета Java Standart Edition

Версия	Количество классов и интерфейсов	Количество методов и полей
1.0	212	2125
1.1	504	5478
1.2	1781	20935

1.3	2130	23901
1.4	3020	32138

Распространенные заблуждения относительно языка Java.

Язык Java – это расширение языка HTML.

Java— это язык программирования, а язык HTML— это способ описания структуры Web-страниц. Между ними нет ничего общего, за исключением расширений языка HTML, позволяющих размещать на Web-страницах апплеты, написанные на языке Java.

Я работаю на языке XML, поэтому мне не нужен язык Java.

Язык Java— это язык программирования, а язык XML— просто способ описания данных. Данные, описанные с помощью языка XML, можно обрабатывать с помощью программ, написанных на любом языке программирования, но лишь библиотека API языка Java содержит превосходные средства поддержки для обработки таких данных. Кроме того, в языке Java реализованы многие возможности языка XML. Более подробно они описаны во втором томе.

Язык Java легко выучить.

Нет ни одного языка программирования, столь же мощного, как язык Java, который можно было бы легко выучить. Игрушечные программы писать довольно просто, однако выполнять серьезную работу всегда трудно. Кроме того, обратите внимание на то, что в этой книге обсуждению собственно языка Java посвящено только четыре главы. В остальных главах объясняется, как работать с его библиотеками, содержащими тысячи классов и интерфейсов, а также многие тысячи функций. К счастью, знать каждую из них не

обязательно, однако для выполнения реального проекта от пользователя требуется на удивление много знаний.

Язык Java со временем станет универсальным языком программирования для всех платформ.

Теоретически это возможно. Именно об этом мечтают все производители программного обеспечения, кроме компании Microsoft. Однако есть много приложений, прекрасно работающих на персональных компьютерах, которые не будут так же хорошо работать на других устройствах или под управлением браузера. Кроме того, эти приложения написаны так, чтобы максимально использовать возможности процессоров и машинозависимых библиотек. В любом случае они уже перенесены на все важные платформы. К таким приложениям относятся текстовые и графические редакторы, а также Web-браузеры. Обычно эти приложения пишутся на языках C или C++, и пользователь ничего не выиграет, если переписать их на языке Java. Помимо всего прочего, после переписывания на языке Java эти программы станут более медленными и менее эффективными, по крайней мере, в ближайшем будущем.

Язык Java — это просто еще один язык программирования.

Java— прекрасный язык программирования. Многие программисты отдают предпочтение именно ему, а не языкам C или C++. Однако в мире существуют сотни прекрасных языков программирования, так никогда и не получивших широкого распространения, в то время как языки с очевидными недостатками, такие как C++ и Visual Basic, достигли ошеломительных успехов.

Почему? Успех любого языка программирования в основном определяется практичностью его системы поддержки (support system), а не элегантностью его синтаксиса. Существуют ли полезные, удобные и стандартные библиотеки, позволяющие сделать именно то, что необходимо программисту? Разработана ли удобная среда для создания и отладки программ? Интегрирован ли язык и его инструментарий в компьютерную инфраструктуру? Язык Java достиг успехов в области серверных приложений, поскольку его библиотеки классов позволяют легко сделать то, что раньше было трудным, например, поддерживать работу в сети и многопоточность. Тот факт, что язык Java уменьшил количество ошибок, связанных с указателями, также говорит в его пользу. Благодаря этому производительность труда программистов повысилась. Однако не в этом кроется причина его успеха.

Программы на языке Java интерпретируются, значит, серьезные приложения будут слишком медленно выполняться на конкретной платформе.

Многие программы затрачивают большую часть времени на взаимодействие с пользовательским интерфейсом или ожидание данных из сети. Все программы, независимо от того, на каком языке они написаны, должны реагировать на щелчок мыши за определенное время. Разумеется, не следует выполнять

задачи, требующие высокой производительности центрального процессора, с помощью интерпретатора языка Java. Однако на платформах, допускающих синхронную компиляцию, нужно лишь запустить байт-коды на выполнение, и большинство вопросов просто отпадут.

В конце концов язык Java отлично подходит для разработки сетевых программ. Опыт показывает, что он легко поддерживает высокую скорость передачи в сети даже во время таких интенсивных вычислений, как шифрование. Поскольку скорость работы на языке Java выше скорости передачи данных в сети, становится неважным, что программы на языке C++ могут работать еще быстрее. На языке Java легче программировать, а программы, написанные на нем, машинезависимы.

Все программы на языке Java выполняются под управлением Web-браузеров.

Все апплеты, написанные на языке Java, действительно выполняются под управлением Web-браузеров. Собственно, это и есть определение апплета— программа, выполняемая Web-браузером. Однако вполне возможно и уместно создавать самостоятельные программы на языке Java, которые выполняются независимо от Web-браузера.

Эти программы (обычно называемые приложениями) являются полностью машинезависимыми. Просто берите программу и выполняйте ее на другой машине! Поскольку язык Java более удобен и менее подвержен ошибкам, чем язык C++, он представляет собой хороший выбор. В сочетании со средствами доступа к базам данных, например, пакетом Java Database Connectivity, язык Java становится просто неотразимым. Особенно удобно его изучать в качестве первого языка программирования.

Большинство программ в этой книге являются абсолютно самостоятельными. Конечно, апплеты очень забавны. Однако приложения на языке Java на практике более важны и полезны.

Апплеты на языке Java представляют собой большую опасность для системы защиты.

Было опубликовано несколько отчетов об ошибках в системе безопасности языка Java. Большинство из них касалось реализации языка Java с помощью конкретного браузера. Исследователи восприняли это как вызов и принялись искать лазейки в системе защиты языка Java, чтобы преодолеть силу и сложность модели безопасности апплетов. Обнаруженные технические ошибки вскоре были исправлены, и, насколько мы знаем, ни одна реальная система никогда не была скомпрометирована. Чтобы оценить важность этого факта, вспомните о миллионах вирусных на выполняемые файлы операционной системы Windows и макросы текстового процессора Word, действительно вызвавшие немало бед, и удивительно беззубую критику слабостей атакованной платформы. Кроме того, механизм ActiveX в браузере Internet Explorer мог бы вызвать много нареканий, однако способы его взлома настолько очевидны, что лишь немногие специалисты потрудились опубликовать свои изыскания.

Некоторые системные администраторы даже стали выключать системы защиты языка Java в своих браузерах, чтобы пользователи могли, как и прежде загружать исполняемые файлы, элементы управления ActiveX и документы, созданные с помощью текстового процессора Word. Забавно, что в настоящее время вероятность преодоления системы защиты апплетов на языке Java сравнима с вероятностью погибнуть в авиационной катастрофе, в то время как риск заразить компьютер, открыв документ, созданный текстовым процессором Word, сравним с риском погибнуть под колесами автомобиля, перебегая дорогу в час пик.

Язык JavaScript — упрощенная версия языка Java.

Язык JavaScript — это язык разработки сценариев, который можно использовать на Web-страницах. Он был разработан компанией Netscape и сначала назывался LiveScript. Синтаксис языка JavaScript напоминает синтаксис языка Java, однако на этом их сходство заканчивается (за исключением имени, конечно). Подмножество языка JavaScript было стандартизовано под именем ECMA-262, однако его расширения, необходимые для реальной работы, стандартизованы не были. В результате программа на языке JavaScript, которую можно было бы выполнять как под управлением браузеров компании Netscape, так и с помощью браузера Internet Explorer, осталась несбыточной мечтой.

Язык Java совершит революцию в области клиент-серверных приложений.

Это возможно. Именно в этой области язык Java особенно полезен. Некоторые серверные приложения, например BEA Weblogic, полностью встроены в язык Java.

Язык Java отменит компонентную модель вычислений.

Вряд ли найдутся хотя бы два человека, у которых совпали бы мнения о компонентах. В качестве визуальных компонентов управления в версии Java 1.1 были предложены функциональные компоненты JavaBeans, аналогичные компонентам ActiveX, использующимся для создания графического пользовательского интерфейса.

Функциональные компоненты JavaBeans позволяют делать то же самое, что и компоненты ActiveX, за исключением того, что они автоматически являются кроссплатформенными. С точки зрения серверных приложений промышленные компоненты, предназначенные для повторного использования, легко применять при разработке широкого круга серверных приложений. Возможно, в будущем возникнет рынок этих компонентов, аналогичный рынку компонентов ActiveX в мире Windows.

ПРОГРАММНАЯ СРЕДА. УСТАНОВКА JAVA НА КОМПЬЮТЕР.

Для того, чтобы начать программировать на языке Java, первоначально нужно установить программную среду на свой компьютер.

1) Скачиваем инсталлятор Java.

Скачать инсталляционный пакет можно на сайте компании Oracle по ссылке:

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

Качаем JDK последней версии

Откройте **Панель управления -> Система** там будет указан Тип системы.

Если тип вашей системы — 32 разрядная, то нужно устанавливать версию для x86, если 64 разрядная, то x64.

Внимание, перед тем как начать скачивание, нужно выбрать пункт Accept License Agreement вверху таблицы.

2) Инсталлируем Java.

Для того, чтобы установить Java на компьютер, запустите скаченный файл, после чего начнется процесс установки

Нажмите Next.

Установка по умолчанию будет происходить в директорию C:\Program Files\Java\jdk1.x.x_xx (вместо иксов номер версии), при желании директорию можно изменить, нажав на кнопку Change. Следуйте инструкциям мастера установки, и через несколько минут установка будет завершена.

3) Настройка переменных сред.

На следующем шаге необходимо задать пути к выполняемым файлам, а именно задать системные переменные PATH, CLASSPATH и JAVA_HOME

Для этого откройте **Панель управления -> Система**, выберите **Дополнительные параметры системы**. В появившемся окне нажмите кнопку **Переменные среды**. В открывшемся окне в блоке **Системные переменные** нужно проверить существует ли переменная **PATH**, если переменная не существует, то ее нужно создать нажав на кнопку **Создать**, если переменная **PATH** уже существует, то ее нужно **Изменить**. В следующем окне для переменной **PATH** нужно установить **Значение переменной** — путь к каталогу, в который была установлена Java. Также необходимо создать переменную **CLASSPATH**, если она еще не создана. В качестве ее значения необходимо указать «.;» Подобным образом создается/изменяется переменная **JAVA_HOME**. Значение этой переменной — путь к каталогу, в который была установлена Java (как для **PATH**), но уже без поддиректории **bin**. Например:

Значение в PATH: *C:\Program Files\Java\jdk1.7.0_25\bin*

Значение в JAVA_HOME: *C:\Program Files\Java\jdk1.7.0_25*

После того, как были прописаны переменные, **установку Java можно считать завершенной!**

4)Проверка установки.

Чтобы проверить, успешно ли прошла установка, откройте командную строку. Для этого в меню Windows **Пуск** в строке поиска введите команду **cmd** и нажмите **Enter**. После этого откроется командное окно, в котором нужно ввести следующую команду.

Java -version

После чего, если отобразится информация об установленной версии java, то установка прошла успешно.

ПЕРВАЯ ПРОГРАММА HELLO WORLD НА JAVA

1) Создание исходного файла

Для начала нужно написать текст программы на языке Java и сохранить его. Это и будет нашим исходным файлом. Для создания исходного файла подойдет любой текстовый редактор, например стандартный «Блокнот». Однако, существуют и другие текстовые редакторы, которые более удобны для написания кода. Можно воспользоваться ,например, **Notepad++**. Это бесплатный текстовый редактор, который поддерживает синтаксис большинства языков программирования, в том числе и Java.

Итак, открываем текстовый редактор и пишем в нем код программы **Hello World**, цель которой — вывод на экран сообщения **Hello World!**

```
Class HelloWorld {  
  
    public static void main(String[] args) {  
  
        // Display “Hello World!”  
    }  
}
```

```
System.out.println("Hello World!");
```

```
}
```

```
}
```

После написания этого кода, файл нужно сохранить под именем HelloWorld.java.

Для этого в вашем текстовом редакторе нужно выбрать пункт меню **Файл-> Сохранить как...** Если вы пользуетесь стандартным Блокнотом Windows, то для того, чтобы сохранить файл с расширением .java необходимо при сохранении выбрать **Тип файла: Все файлы** и ввести **Имя файла: HelloWorld.java** **Будьте внимательны!** Файл должен называться в точности так, как называется наш класс — HelloWorld. Так же важно учитывать регистр букв. HelloWorld и helloworldv данном случае это разные слова!

Обратите также **внимание на кодировку** в которой сохраняете файл. Должно быть выбрано **ANSI**.

2) Компиляция исходного файла

Исходный файл с кодом программы создан, теперь перейдем к компиляции. Для компиляции Java предназначен компилятор javac, который входит в состав установленного нами пакета JDK.

Для того, чтобы скомпилировать исходный файл, открываем командную строку. Для этого в меню Windows Пуск в строке поиска вводим команду cmd и жмем Enter. После этого откроется командное окно.

Теперь в нем нужно изменить текущий каталог на тот, в котором находится наш исходный файл (например C:\studyjava\). Для этого вводим следующую команду:

```
cd C:\studyjava\
```

и нажимаем Enter.

После того, как директория изменилась, вводим команду компиляции

```
javac HelloWorld.java
```

нажимаем Enter.

То есть, мы не получим никакого подтверждения, о том, что программа скомпилировалась успешно. Однако, в папке с нашим исходным файлом, должен появиться файл HelloWorld.class. Это можно проверить с помощью команды

```
dir
```

Эта команда выводит на экран список всех файлов, находящихся в выбранной директории

Если файл HelloWorld.class присутствует в этом списке, то это значит, что программа скомпилировалась успешно.

Если в коде программы есть ошибка, то компилятор Java при компиляции нам об этом сообщит.

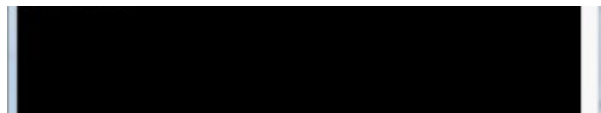
3) Запуск программы

Переходим к последней стадии — запуску программы.

Вводим в командном окне:

```
java HelloWorld
```

и если все перед этим было сделано правильно, то получаем результат — вывод сообщения «Hello World!»



Конечно, для написания, компилирования и запуска программ на языке Java существуют более удобный инструмент, нежели Блокнот и командная строка. Этот так называемая Интегрированная среда обработки IDE.

Обзор IDE средств для программирования на Java.

За более чем десятилетнюю историю языка Java, сменилось ни одно поколение интегрированных сред разработки (Integrated Development Environment – IDE. Эволюция IDE средств обусловлена множеством факторов, совокупность которых называется – информационные технологии, включающих в себя программную и аппаратную составляющие

И если первые IDE средства, представляли в сегодняшнем понимании примитивный текстовый редактор, который служил только для набора исходных кодов, а всю остальную работу, начиная от компиляции и заканчивая окончательной сборкой проекта приходилось выполнять вручную, то сегодняшние современные, являются фактически многофункциональными устройствами, берущими на себя не только привычные функции но и ряд дополнительных начиная от автоматического создания Javadoc, рефакторинга, профайлинга, UML-проектирования, клиента для подключения к любой СУБД и заканчивая такими экзотическими, не относящимися к процессу разработки, как проверка орфографии.

А начинающим программистам трудно оценить адекватно то или иное средство, поэтому, как правило, его самостоятельный выбор происходит по двум критериям – интуитивная понятность интерфейса и различные временно – ресурсные характеристики, такие как загрузка, компиляция, запуск, занимаемый объем ОЗУ. Ведь на первых порах необходимы только компилятор, отладчик, и непосредственно сама Java-машина

Список IDE для Java.

№	Название	Производитель	Сайт производителя
1	Applet Designer Professional	TVObjects	http://www.tvobjects.com/
2	Apptivity	Progress Software	http://apptivity.progress.com/
3	Awesome Jawa 1.0	Awesume Interactive Design	http://www.awesome.se/en/index.htm
4	Bluette	free RAD Java tool	http://blue.donga.ac.kr/bluette/
5	Bongo	Marimba	http://www.marimba.com/
6	Clarion Internet Edition	TopSpeed Corporation	http://www.topspeed.com/
7	CodeWarrior Pro	MetroWerks	http://www.metrowerks.com/desktop/pro/
8	Cosmo Code	SGI	http://www.sgi.com/Products/cosmo/code/index.html
9	ED for Windows	Soft As It Gets	http://www.getsoft.com/ed_java.html
10	Elixir	Elixir Technology Pte Ltd	http://www.elixir.com.sg/
11	Free Builder	FreeBuilder collective	http://www.freebuilder.com/
12	GRASP	GRASP Project	http://www.eng.auburn.edu/grasp/
13	Grinder	Paradigm Exchange	http://www.tpex.com/features.htm
14	Hyperwire	Kinetix	http://www.ktx.com/
15	Jamba Animator	Interleaf	http://www.jamba.com/
16	Jamba	Interleaf	http://www.jamba.com/
17	JavaMan	HartWare	http://homepage.dave-world.net/~hardware/
18	Java Studio	Sun Microsystems	http://www.sun.com/
19	Javelin	Step Ahead	http://www.ozemail.com.au/~stepsoft/
20	JaWiz	InfinityEdge Systems	http://www.infinityedge.com/
21	Jbuilder	Inprise	http://www.inprise.com/jbuilder/
22	JdesignerPro	BulletProof	http://www.bulletproof.com/
23	JDE for Emacs	Paul Kinnucan	http://sunsite.auc.dk/jde/
24	JADI Sujal	Shah	http://dan.hcf.jhu.edu/sujal/winjadi/
25	JIG	S Cubed	http://www.scubed.cc/
26	Jipe	Envision Internet Services	http://www.users.globalnet.co.uk/~eis/jipe.htm
27	Jpad	ModelWorks	http://www.modelworks.com/
28	Kalimantan	Real-Time	http://www.real-time.com/java/kalimantan/index.html

		Enterprises, Inc.	
29	KAWA	TEK-TOOLS, Inc.	http://www.tek-tools.com/kawa/
30	Lava	Dan Page	http://www.hnet.demon.co.uk/products/lava/index.html
31	Lemur	Island Design	http://www.island-design.co.uk/
32	NetBeans	NetBeans, Inc.	http://www.netbeans.com/
33	OEW 1.0	Innovative Software	http://www.isg.de/OEW/Java/
34	PARTS for Java	ObjectShare	http://www.objectshare.com/p4j/p4j2info.htm
35	PowerJ	Sybase	http://www.sybase.com/products/powerj/
36	Roaster	Natural Intelligence	http://www.roaster.com/roaster/
37	SniFF+	TakeFive Software	http://www.takefive.com/sniff/
38	Spirit	eVisNet Limited	http://www.evis.net/
39	SuperCede	SuperCede, Inc.	http://www.supercede.com/
40	Visaj	Imperial Software Technology	http://www.ist.co.uk/
41	Vision Jade	Vision Software	http://www.vision-soft.com/
42	VisualAge for Java	IBM	http://www.software.ibm.com/ad/vajava/
43	Visual Cafe	Symantec	http://cafe.symantec.com/
44	Visual J++	Microsoft	http://www.microsoft.com/visualj/
45	Web Application Platform	SilverStream	http://www.silverstream.com/products/main/main_f.htm
46	Java WebIDE	Chami.com	http://www.chami.com/webide/
47	WipeOut	softwarebuero m&b	http://www.softwarebuero.de/wipeout-eng.html
48	IntelliJ IDEA	IntelliJ IDEA	http://www.jetbrains.com/idea/
49	Eclipse	eclipse.org	http://www.eclipse.org
50	Jdeveloper	Oracle	http://www.oracle.com/technology/products/jdev/index.html
51	Jcreator	Xinox Software	http://www.jcreator.com/
52	jEdit	jEdit	http://www.jedit.org/
53	X-Develop	Omnicores Software	http://www.omnicore.com/
54	Gel IDE	Gexperts Inc	http://www.gexperts.com/
55	IBM Web Sphere Studio Application Developer	IBM Software	http://www-306.ibm.com/software/awdtools/studioappdev/
56	eXtend	NOVELL	http://www.novell.com/products/extend/

Проклассифицируем IDE средства, для этого определим классификационные признаки, по которым IDE-средства можно разбить на группы.

Во первых, способ распространения, соответственно IDE можно разделить на платные (Jbuilder, Visual Cafe...) и бесплатные (NetBeans, Eclipse, Gel), которые любой желающий может скачать с сайта производителя.

Во-вторых, важно наличие визуального построителя интерфейса, что в свою очередь выдвигает определенные требования к ресурсам.

По этому критерию можно выделить средства, которые не имеют визуальных средств а для разработки GUI (простые), например Gel, Jcreator и которые имеют (сложные) – Jbuilder, Idea.

Протестируем некоторые из IDE-средств:

Результаты тестирования первых IDE – средств.

Параметры	Gel	Jcreator 3.5	Jbuilder X	Idea 5	NetBeans 4.1
Время установки	1 сек	1-1,5 сек	58 сек	22 сек	50сек
размер установленного пакета	10,3 Мб	6,49 Мб	332 Мб	157 Мб	118 Мб
Размер дистрибутива	4,21 Мб	3,6 Мб	178 Мб	51,6 Мб	46,4Мб
Запуск	5с < t < 6с	1-1,5 сек	9 сек	6 сек	6 сек
Занимаемый объем в ОЗУ	17828 Кб	12960 Кб	74128 Кб	65604 Кб	61409 Кб
Занимаемый объем виртуальной памяти	9020 Кб	14960 Кб	78588 Кб	76416 Кб	70406 Кб

* За базовую конфигурацию была взята машина на базе: CPU Athlon64 3000+; ОЗУ – 1024Мб DDR-SDRAM (pc3200)

В результате чего будет установлено, что на начальном этапе работы, когда идет процесс изучения синтаксических особенностей языка, нет необходимости в использовании тех возможностей, которые представляют сложные IDE, хотя, безусловно, многих заинтересовали способы быстрой разработки интерфейса, так как эта возможность наиболее интуитивно понятная, и сразу позволяет получать результаты. Исходя из этого, проанализируем, с каких позиций программист подходит к самостоятельному выбору средства.

1. **Интерфейс.** Этот первый компонент, с которым сталкивается пользователь после установки и который формирует первое впечатление о программе, и на основании которого может быть осуществлен окончательный выбор.
2. **Настройка.** Соответственно после установки и первого запуска среды разработки выполняется ее настройка, т.е. указываются пути, по которым располагаются установленные SDK, DOCS, J2EE. В этом компоненте, как правило, существенных различий не наблюдается. Более того, современные средства, как правило, самостоятельно определяют установленные компоненты.
3. **Редактор кода.** Настройка отображения исходных кодов, как правило, тоже не отличается разнообразием, в любом средстве легко можно настроить кегль и ее размер, а так же цвет. Немаловажным преимуществом является наличие помощника когда, при «зависании» мышки на любой переменной или методе всплывает довольно подробный ToolTip (контекстное окошко) в котором развернуто, описаны все параметры объекта. Также есть масса приятных мелочей, вывод нумерации строк, отображение структуры класса, показ символов абзаца, проверка орфографии.

Проанализировав, по этим признакам, все существующие на сегодняшний день IDE – средства, можно заметить, что существенных различий кроме дизайна интерфейса между ними нет и очевидно понятие удобства является субъективным.

Рейтинг Java-IDE,
используемых в настоящее время

№	Название	% голосов	Оценка
1	Eclipse	19.77	4.6
2	IntelliJ IDEA	19.06	4.7
3	NetBeans	7.11	4.1
4	Jbuilder	5.68	4.2
5	Jdeveloper	2.13	4.0
6	Jcreator	1.70	3.9

Таким образом, на сегодняшний день лидерами среди IDE-средств разработки на Java являются IntelliJ IDEA, Eclipse и NetBeans. В целом все они имеют примерно одинаковые функциональные возможности, и довольно трудно оценить какое из них является лучшим. Поэтому при выборе средства, очевидно, нужно ориентироваться на IDE, на котором ведется разработка проекта, если программист вливается в команду разработчиков, или выбирать наиболее доступное, если планируется разработка самостоятельного проекта.

Скачать Eclipse можно пройдя по ссылке: http://fresoft.ru/eclipse_ide. Также, если писать программы на Java в Eclipse, требуется утилитарная работа по написанию gui-приложений. Решение этой проблемы есть – это **WindowBuilder** – визуальный дизайнер интерфейсов на Swing, GWT и SWT. Для платформы Eclipse выполнен в виде плагина. Некогда WindowsBuilder принадлежал компании Instntiations, но сравнительно недавно (на момент написания статьи) компания Google выкупила часть данной компании, которая занималась Java-разработками. На данный момент WindowsBuilder распространяется свободно.

Порядок установки WindowBuilder:

1 Распаковать архив

WB v1.6.1 UpdateSite for Eclipse4.
3.zip

2 Запустить Eclipse

3 Зайти в меню Eclipse

меню\Help\Install new software... ->

Откроется диалог “Install”

4 В диалоге “Install” нажать кнопку

«Add» -> Откроется диалог “Add

Repository”

5 В диалоге “Add Repository” нажать

кнопку “Local...”¹⁷

6 В открывшемся диалоге выбрать

распаковали архив
WB v1.6.1 UpdateSite for Eclipse4.3.zip. Нажать кнопку ОК -> диалог закроется

7 В диалоге “Add Repository” в поле Name введите “WindowBuilder”, после чего нажмите кнопку ОК -> диалог закроется

8 В диалоге “Install” в поле Name поставьте галочки для всех компонентов – Swing Designer, SWT Designer, WindowBuilder Engine (можно нажать Select All)

Самые популярные языки программирования (в России):

9 Нажмите кнопку Next
<http://itmozg.ru/news/1233/#.UyfhkIWdHGo> (конец 2014)

10 И далее до победного конца!

Индекс TIOBE (2014)

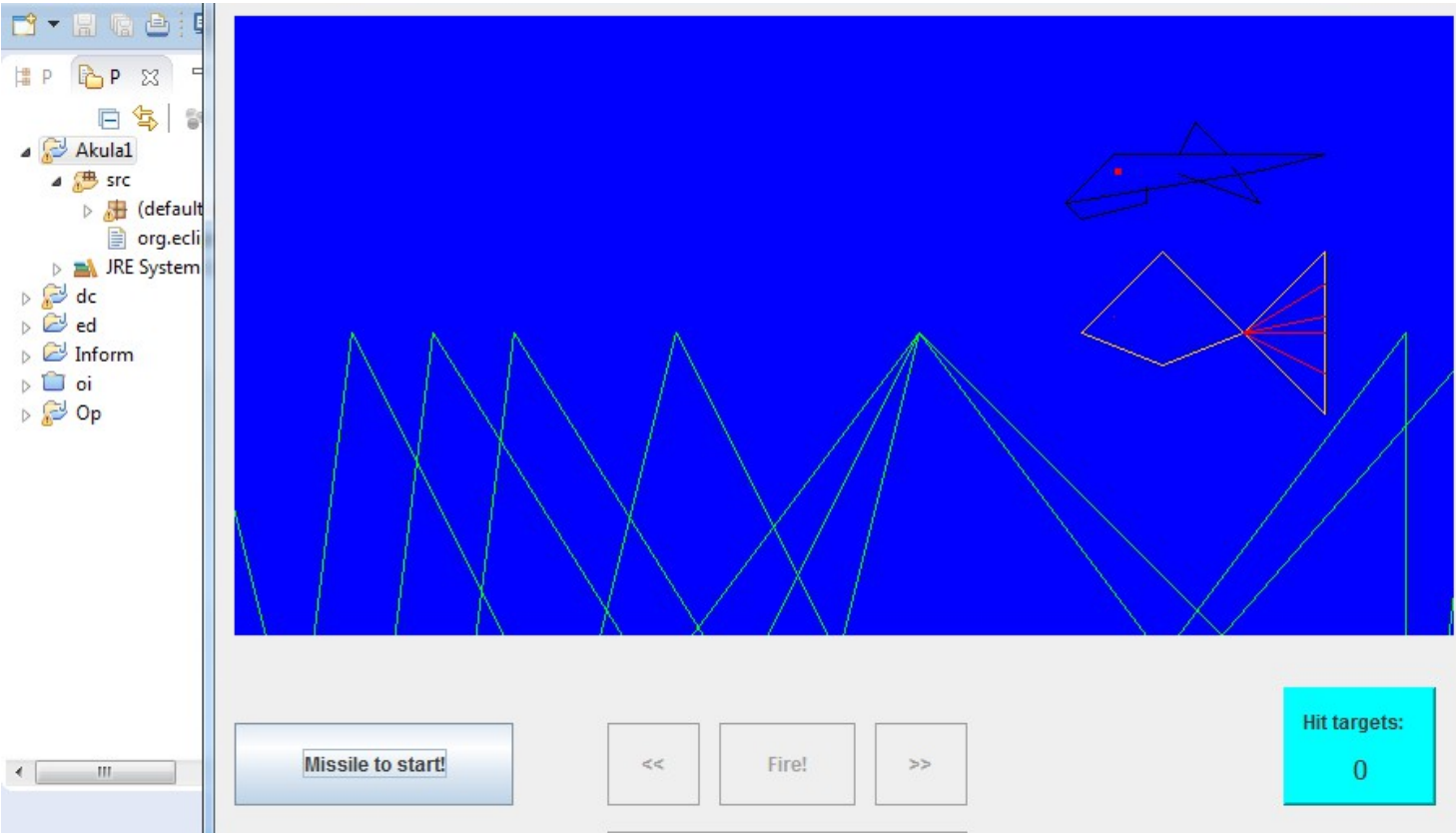
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

Игра, созданная мной на Java.

Aircraft – класс самолета.

Missile – класс зенитной ракеты

AirPanel – панель с отрисованными на ней самолетом и ракетой



Список используемой литературы:

1. Жмайлов Б.Б. Преимущества и недостатки разработки Java-программ без использования IDE-средств. Журнал "Вестник компьютерных и информационных технологий" №6, 2006г.
2. http://www.javaportal.ru/java/ide/oracle_JDeveloper10g.html, Сергей Бердачук, "Oracle JDeveloper 10g - Среда разработки Java приложений"
3. http://www.javaportal.ru/java/ide/intro_eclipse.html, Алексей Литвинюк, "Введение в интегрированную среду разработки Eclipse"
4. <http://www.javaportal.ru/projects/taidej/reviewJBuilder.html>, Александр Демьяненко, "Краткий обзор IDE – Jbuilder"
5. <http://www.javaportal.ru/projects/taidej/reviewIdea.html>, Александр Демьяненко, "Краткий обзор IDE – Idea"
6. <http://www.javaportal.ru/projects/taidej/reviewGel.html>, Борис Жмайлов, "Краткий обзор IDE - Gel"

7. http://www.javaportal.ru/java/ide/review_refactoring.html, Обзор средств автоматизированного рефакторинга в Java IDE