Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

**Реферат**

По дисциплине «Традиционные и интеллектуальные информационные технологии»

Тема: «Язык программирования Java»

**Выполнил:**

Студент 1 курса

Группы ИИ-21

Литвинюк Т. В.

**Проверил:**

Анфилец С. В.

Брест 2021

**СОДЕРЖАНИЕ**

**ВВЕДЕНИЕ 3**

1. **ИСТОРИЯ СОЗДАНИЯ ЯЗЫКА ПРОГРАМИРОВАНИЯ JAVA 4**
2. **СФЕРЫ ПРИМЕНЕНИЯ ЯЗЫКА JAVA 6**
3. **ПЛЮСЫ И МИНУСЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ JAVA 9**

**ЗАКЛЮЧЕНИЕ 15**

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 16**

**ВВЕДЕНИЕ**

 Современную жизнь общества трудно представить без компьютеров. Большинство из нас уже не может существовать ни одного дня без них. Эти умные машины полностью заполонили нашу повседневную жизнь. Отправить электронное письмо, набрать и распечатать текст, прочитать новости в интернете, скачать фильм, музыку и т. д. ⎯ это всё люди делают с помощью компьютеров. А что касается трудоустройства? Чаще всего, современные вакансии связывают с навыками работы на компьютере. Знание языка программирования Java ⎯ одно из требований при трудоустройстве для IT-специалистов. Что это за язык программирования, владение которым так нужно работодателям? Это наиболее часто используемый язык программирования в мире. На Java работают 3 миллиарда устройств. Индекс сообщества программистов обновляется ежемесячно и регулярно указывает Java как наиболее популярный язык программирования. Чем обеспечена такая стабильно высокая популярность? Потому что существует много реальных сфер применения. Java занимает второе место среди лучших языков программирования. Благодаря этому, можно говорить об актуальности данной темы.

**Предмет реферата** — язык программирования Java.

**Цель реферата:** обосновать актуальность языка программирования Java.

**Задачи реферата:**

* изучить историю создания языка программирования Java;
* изучить сферы применения языка Java;
* изучить плюсы и минусы Java.

1. **ИСТОРИЯ СОЗДАНИЯ ЯЗЫКА ПРОГРАМИРОВАНИЯ JAVA**

В 1990 году разработчик ПО компании Sun Microsystems Патрик Нотон понял, что ему надоело поддерживать сотни различных интерфейсов программ, используемых в компании, и сообщил исполнительному директору Sun Microsystems Скотту МакНили о намерении перейти работать в компанию NeXT. МакНили попросил Нотона рассказать о причинах недовольства. Нотон предоставил свои претензии, раскритиковав недостатки Sun Microsystems, в том числе разрабатываемую архитектуру ПО NeWS. К удивлению Нотона, его письмо было разослано всем ведущим инженерам Sun Microsystems, которые не замедлили откликнуться и высказать горячую поддержку своему коллеге и одобрение его взглядов на ситуацию в Sun Microsystems. Обращение вызвало одобрение также и у высшего руководства компании Билла Джоя, основателя Sun Microsystems, и Джеймса Гослинга, начальника Нотона. В день, когда Нотон должен был уйти из компании, было принято решение о создании команды ведущих разработчиков, чтобы они делали что угодно, но в итоге, придумали что-то необычное. Команда из шести человек с кодовым названием Green погрузилась в исследование бытовых устройств, таких как Nintendo Game Boys, устройств дистанционного управления. Команда Green пыталась найти средство, с помощью которого можно было бы установить взаимодействие между этими устройствами. Такие электроприборы, как видеомагнитофоны, проигрыватели лазерных дисков, стереосистемы - все они были реализованы на разных процессорах. Это означало, что, если производитель захочет добавить телевизору или видеомагнитофону 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, который рос под окном Гослинга. Компания Sun Microsystems переименовала команду Green в компанию First Person. Новая компания обладала другой концепцией, но не могла найти ей нужного применения. После ряда неудач неожиданно ситуация для компании резко изменилась: был анонсирован Mosaic - так родился World Wide Web, с которого началось бурное развитие Internet. Нотон предложил использовать Oak в создании Internet-приложений. Так Oak стал самостоятельным продуктом. В ближайшем будущем был написан Oak-компилятор и Oak-браузер "WebRunner". В 1995 году компания Sun Microsystems 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-аплеты - небольшие программы, загружаемые через Internet. WebRunner был переименован в HotJava и компания Netscape встала на поддержку Java-продуктов. Но официальной датой создания языка Java считается 23 мая 1995 года, после выпуска компанией Sun первой реализации Java 1.0. Она гарантировала «Напиши один раз, запускай везде», обеспечивая недорогой стоимостью на популярных платформах. 13 ноября 2006 года Sun выпустила большую часть как свободное и открытое программное обеспечение в соответствии с условиями GNU General Public License (GPL). После 8 мая 2007 года судьба Java сложилась иначе. Компания завершила процесс, делая все, чтобы исходный код был бесплатным и открытым, кроме небольшой части кода, на который компания не имела авторских прав.

1. **СФЕРЫ ПРИМЕНЕНИЯ ЯЗЫКА JAVA**

Java — это не только язык программирования, но и экосистема инструментов, охватывающая почти все, что может понадобится при программировании на Java. В нее входят:

* **Java Development Kit (JDK) —** комплект разработчика Java. С помощью JDK и стандартного блокнота можно писать и запускать / компилировать код на Java.
* **Java Runtime Environment (JRE)** — исполняющая система Java. Механизм распространения программного обеспечения, состоит из автономной виртуальной машины Java, стандартной библиотеки Java (Java Class Library) и инструментов настройки.
* **Integrated Development Environment (IDE)** —интегрированная среда разработки. Инструменты, которые запускать, редактировать и компилировать код. Самые популярные из них — IntelliJ IDEA, Eclipse и NetBeans.

Java используется во многих сферах, начиная с e-commerce веб-сайтов и заканчивая приложениями под Android, в научных исследованиях и финансах, игах вроде Minecraft и настольных приложениях Eclipse, NetBeans и IntelliJ IDEA.

**Мобильные приложения Android**

Java Micro Edition — это популярная кроссплатформенная среда, применяемая для создания приложений смартфонов. Примеры мобильных приложений, написанных на Java: Netflix, Tinder, Google Earth, Uber и т. д.

**Игры на Java**

Сайт Java-Gaming.org полностью посвящен созданию игр на Java. Библиотека LWJGL значительно упрощает этот процесс

**Серверные финансовые приложения**

Глобальные инвестиционные банки Goldman Sachs, Citigroup, Barclays, Standard Charted используют Java для написания фронт и бэк-офисов систем электронной торговли, систем расчетов и подтверждения транзакций, обработки данных и т. п.

**Веб-приложения на Java**

Из представленной ниже таблицы следует, что Java — самый популярный серверный язык среди ведущих технологических компаний. В сравнении с другими программирования, Java обладает лучшей безопасностью, адаптируемостью, и масштабируемостью приложений.

На Java создано множество opensourse-инструментов для мониторинга и обновления данных. К примеру, Netflix использует Java одновременно для разработки инфраструктуры на Linux и кодирования видео на Windows-машинах. Веб-приложения электронной коммерции применяют Java на таких платформах, как Broadleaf.

**Программы для сопровождения кода**

Инструментальное программное обеспечения — это набор компьютерных программ, которые разработчики используют для программирования, анализа, обслуживания, отладки и поддержки других программ и приложений. Примеры: Eclipse, IntelliJ IDEA, IDE Netbeans.

**Встраиваемые системы на Java**

Первоначально язык Java был разработан для встраиваемых устройств. Девизом Java было «Напиши один раз – запускай где угодно». В результате Java стал одним из лучших языков для разработки IT-приложений. Встроенные модули Java предоставляют различные способы взаимодействия с сервером, а архитектура модели безопасности песочницы обеспечивают высокий уровень защиты данных. Модули Cinterion Java успешно используются в среде межмашинного взаимодействия (M2M-machine to machine), включая умные счетчики, системы управления автопарком, промышленной автоматизации и мобильного здравоохранения.

**Технологии больших данных**

Большие данные — одна из самых популярных тем в мире новейших технологий. Автоматическая сборка мусора и надежное управление памятью делают Java одним из приоритетных языков для реализации алгоритмов машинного обучения. Многие известные средства обработки больших данных (Hadoop, Spark, Mahout) являются Java-фреймворками.

**Примеры использования Java в крупных корпорациях**

Компания Square помогает продавцам превращать мобильные устройства в платежные системы. Программное обеспечение Square построено с применением Ruby и C++, но стек обработки денежных транзакций основан на Java. Java — действительно единственный выбор, когда дело доходит до требований для такой компании, как наша — с высокими требованиями к производительности и экстремальным требованиям к масштабируемости. Альтернативы нет. Целиком написанная на Java программная среда Hadoop используется для надежного и масштабного распределения вычислительных систем.

Java предлагает хороший компромисс между производительностью разработчика и производительностью во время выполнения. Разработчики получают преимущества простого, мощного, безопасного языка с широким набором высококачественных библиотек.

Для [Twitter](https://en.wikipedia.org/wiki/Twitter) нет ничего важнее масштабируемости. Приложением пользуются более 500 млн человек. Проект начинался, как демонстрация того, на что способен фреймворк Ruby on Rails. К сожалению, со временем стало понятно, что RoR плохо справляется с обработкой большого числа запросов данных в реальном времени, поиском по определенной теме или хештегам. Одним из преимуществ Java является масштабируемость – это послужило одной из главных причин перехода с Ruby на Java.

1. **ПЛЮСЫ И МИНУСЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ JAVA**

**Плюсы программирования на Java**

Java — [уже не](https://techcrunch.com/2017/05/17/google-makes-kotlin-a-first-class-language-for-writing-android-apps/) единственный официально поддерживаемый язык для разработки на Android. Java далеко не единственный выбор в веб-программировании. Тем не менее, Java идет в ногу со временем. Давайте рассмотрим, какие преимущества предлагает Java.

* **Объектно-ориентированное программирование**

Java включает в себя **объектно-ориентированное программирование** (OOP) — концепцию, в которой вы не только определяете тип данных и его структуру, но и набор функций, применяемых к нему. Таким образом, структура данных становится объектом, которым можно управлять для создания отношений между различными объектами.

При другом подходе — **процедурном программировании —** нужно следовать четким инструкциям, использовать переменные и функции. При ООП можно группировать эти переменные и функции посредством контекста, маркировать их и ссылаться на функции в контексте каждого конкретного объекта.

Сравнение процедурного и объектно-ориентированного программирования

В чем плюсы ООП?

* При ООП можно повторно использовать объекты в других программах
* ООП предотвращает ошибки, поскольку объекты скрывают информацию, к которой не должно быть доступа
* ООП более эффективно организует структуру программ, в том числе больших
* ООП упрощает обслуживание и модернизацию старого кода
* **Java — язык высокого уровня с простым синтаксисом и плавной кривой обучения**

Java — это язык высокого уровня, то есть он похож на человеческий язык. В отличие от языков низкого уровня, которые напоминают машинный код. Языки высокого уровня преобразуется с помощью компиляторов или интерпретаторов. Это упрощает разработку, делая язык более легким для написания, чтения и обслуживания.

Синтаксис Java основан на C ++, поэтому Java похожа на C. Тем не менее, синтаксис Java проще, что позволяет новичкам быстрее учиться и эффективнее использовать код для достижения конкретных результатов.

Java не так дружелюбен к новичкам, как Python, однако довольно прост для любого разработчика с базовым пониманием фреймворков, пакетов, классов и объектов. Он прост, типизирован и предсказуем, что позволяет учиться мыслить в правильном направлении. Кроме того, новичок всегда может обратиться к множеству бесплатных онлайн-уроков и курсов.

* **Стандарт для корпоративных вычислительных систем**

Корпоративные приложения — главное преимущество Java с 90-х годов, когда организации начали искать надежные инструменты программирования не на C. Java поддерживает множество библиотек — строительных блоков любой корпоративной системы. Библиотеки помогают разработчикам создавать любые функции, которые могут понадобиться компании. Java широко распространен — это язык, который преподают в рамках введения в программирование в большинстве школ и университетов. Возможности интеграции Java впечатляют: большинство хостинг-провайдеров поддерживают Java. Более того, Java — язык, дешевый в обслуживании: работать с Java можно с любого компьютера, вне зависимости от конкретной аппаратной инфраструктуры.

* **Безопасность**

Существует мнение, что Java — безопасный язык, однако это не совсем так. Сам язык не защищает вас от уязвимостей, но некоторые его функции устраняют распространенные уязвимости. Во-первых, в отличие от C, в Java нет указателей. Указатель — это объект, который сохраняет адрес ячейки памяти другого значения, что может вызвать несанкционированный доступ к памяти. Во-вторых, в Java есть **Security Manager**, созданная для каждого приложения политика безопасности, в которой можно указать правила доступа. Это позволяет запускать приложения Java в «песочнице» и устранять таким образом уязвимости.

* **Независимость от платформы («Написать один раз и использовать везде»)**

«Написать один раз и использовать везде» (WORA) — популярная в IT-сфере фраза, с помощью которой Sun Microsystems описывает кросс-платформенные возможности Java. Можно создать Java-приложение на Windows, скомпилировать его в байт-код и запустить его на любой другой платформе, поддерживающей виртуальную машину Java (JVM). Таким образом, JVM служит уровнем абстракции между кодом и оборудованием.

Как работает WORA на Java

Все основные операционные системы, включая Windows, Mac OS и Linux, поддерживают JVM. Если ваша программа не опирается на специфичные для платформы функции и пользовательский интерфейс, ее можно с легкостью перенести: по крайней мере, большую ее часть.

* **Язык для распределенного программирования и комфортной удаленной совместной работы**

Java создавался как язык для распределенного программирования: он имеет встроенный механизм совместного использования данных и программ несколькими компьютерами, что повышает производительность и эффективность труда.

Сравнение распределенного и параллельного программирования

В других языках нужно использовать внешний API для дистрибуции. В Java эта технология встроена. Специфическая для Java методология распределенных вычислений называется **Remote Method Invocation** (RMI). RMI позволяет использовать все преимущества Java: безопасность, независимость от платформы и объектно-ориентированное программирование для распределенных вычислений. Кроме того, Java также поддерживает программирование сокетов и методологию распределения CORBA для обмена объектами между программами, написанными на разных языках.

* **Автоматическое управление памятью**

Разработчикам Java не нужно вручную писать код для управления памятью благодаря автоматическому управлению памятью (AMM). AMM также используется в языке программирования Swift и при очистке памяти приложениями, которые автоматически обрабатывают распределение и освобождение памяти. Что именно это означает?

Эффективность программы напрямую связана с памятью. При этом объем памяти ограничен. При написании приложения на языках с ручным управлением памятью, разработчики рискуют забыть выделить память, что приведет к увеличению объема занимаемой приложением памяти и проблемам с производительностью. Программы очистки памяти ищут объекты, которые больше не используются программой, и удаляют их. Это влияет на работу процессора, однако умная оптимизация и настройка позволяют снизить это влияние.

* **Многопоточность**

Поток — наименьшая единица обработки в программировании. Чтобы максимально эффективно использовать время процессора, Java позволяет запускать потоки одновременно, что называется многопоточностью.

Потоки используют одну и ту же область памяти, поэтому между ними можно быстро переключаться. Потоки независимы друг от друга: один поток не влияет на работу других потоков. Это особенно полезно в играх и программах с большим объемом анимации.

* **Стабильность и сообщество**

Уже много лет развитию Java способствуют сообщество, поддержка Oracle и изобилие приложений и языков на JVM. Кроме того, постоянно выпускаются новые версии Java с новыми интересными функциями.

Сообщество разработчиков Java не имеет себе равных. Около 45% респондентов опроса StackOverflow 2018 используют Java. У Java чрезвычайно большая экосистема хорошо протестированных библиотек и фреймворков для любых задач. Начинающий разработчик, скорее всего, выберет Java: на тему Java-программирования существует более 1000 курсов на Udemy и более 300 на Coursera.

**Минусы программирования на Java**

Рассмотрим недостатки Java-программирования.

* **Платное коммерческое использование**

Недавно Oracle объявила, что с 2019 года компания начнет взимать плату за использование Java Standard Edition 8 в «коммерческих целях». За все новые обновления и исправления ошибок придется заплатить. Плата зависит от количества пользователей или компьютеров.

Текущая версия Java бесплатна для простого использования. Таким образом, каждая использующая Java компания должна оценить, насколько эффективно она использует Java. Компания должна понять, что выгоднее: искать альтернативное решение или продолжать пользоваться Java.

* **Низкая производительность**

У любого языка высокого уровня довольно низкая производительность из-за компиляции и абстракции с помощью виртуальной машины. Однако это не единственная причина низкой скорости Java. Например, приложение очистки памяти: это полезная функция, которая, к сожалению, приводит к значительным проблемам с производительностью, если требует больше 20 процентов времени процессора. Плохая настройка кэширования может вызвать чрезмерное использование памяти. Существует также взаимная блокировка потоков: так происходит, когда несколько потоков пытаются получить доступ к одному и тому же ресурсу. В этом случае происходит кошмар каждого Java-разработчика — ошибка из-за нехватки памяти. Тем не менее умелое планирование может решить все эти проблемы.

* **Отсутствие нативного дизайна**

Для создания графического интерфейса пользователя (GUI) разработчики используют различные инструменты, ориентированные для конкретного языка. Для Android-приложений есть Android Studio, которая помогает создавать приложения с нативным дизайном. Однако, когда дело доходит до пользовательского интерфейса на ПК, Java-инструмента для создания нативного дизайна нет.

Есть несколько инструментов для разработки GUI для Java: самые популярные из них — Swing, SWT, JavaFX, JSF. Библиотека Swing — это старый, но надежный кросс-платформенный инструмент, интегрированный в различные Java-IDE, в том числе Eclipse и NetBeans. Однако, если вы не используете шаблоны, вы заметите несоответствия интерфейса. SWT использует собственные компоненты, но не подходит для сложного интерфейса. JavaFX — лаконичный и современный, но слишком новый. В целом, перед созданием GUI на Java нужно подробнее изучить инструменты.

* **Многословный и сложный код**

Многословность кода может показаться преимуществом, которое поможет при изучении языка. Однако, длинные, чрезмерно сложные предложения затрудняют чтение и просмотр кода. Как и естественные языки, многие языки программирования высокого уровня содержат лишнюю информацию. Java — это более легкая версия неприступного C++, которая вынуждает программистов прописывать свои действия словами из английского языка. Это делает язык более понятным для неспециалистов, но менее компактным.

Сравним Java и Python и увидим, в чем преимущество лаконичного кода Python. В Python не используются точка с запятой, круглые и фигурные скобки. Вместо «и», «или» и «нет» в качестве операторов используются «&&», «||» и «!».

**ЗАКЛЮЧЕНИЕ**

Главная черта Java — адаптивность. Программное обеспечение, написанное на одной платформе, запустится и на других устройствах. Программы Java работают на Java Virtual Machine, предоставляющей оптимизацию кода и управление памятью. Java также обеспечивает масштабируемость — одновременную обработку нескольких запросов и многопоточное выполнение программы. Одним из преимуществ языка Java является возможность переноса программ из одной системы в другую.

Существует множество единомышленников Java программистов. Поскольку язык существует давно, то есть много практик Java. Этот язык самый длительный среди всех остальных, на нем завязаны многие сферы, поэтому он еще долгое время будет актуальным. Изучить Java всегда есть смысл.

Широкие возможности Java, простота применения, независимость от платформы и встроенные функции защиты делают этот язык программирования одним из лучших. Теперь стало понятно, почему работодатели хотят видеть у себя специалистов, владеющих языком программирования Java.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Плюсы и минусы программирования на Java [Электронный ресурс] — Режим доступа: https://www.altexsoft.com/blog/engineering/pros-and-cons-of-java-programming/ — Дата доступа: 20.11.2021.
2. Язык программирования Java: особенности, популярность, ситуация на рынке труда [Электронный ресурс] — Режим доступа: https://ru.hexlet.io/blog/posts/yazyk-programmirovaniya-java-osobennosti-populyarnost-situatsiya-na-rynke-truda — Дата доступа: 20.11.2021.
3. История создания языка программирования Java [Электронный ресурс] — Режим доступа: https://web.informatics.ru/works/17-18/web\_online/barabanov\_n\_v/language\_java.html — Дата доступа: 20.11.2021.
4. Дэвид Уитни. Учимся создавать сайты, приложения и игры / Дэвид Уитни — СПб.: ПИТЕР, 2018. — 207 с.
5. Java [Электронный ресурс] — Режим доступа: https://ru.wikipedia.org/wiki/Java — Дата доступа: 20.11.2021.