

ТЗ на "Специализацию Java"

Иван Игоревич Овчинников

2022-08-13 (02:19)

Оглавление

1	Java Core	2
1.1	Платформа: история и окружение	2
1.2	Управление проектом: сборщики проектов	7
1.3	Специализация: данные и функции	7
1.4	Специализация: ООП	7
1.5	Специализация: Тонкости работы	8
2	Java Development Kit	9
2.1	Исключения	9
2.2	Интерфейсы	9
2.3	Обобщённое программирование	9
2.4	Коллекции	9
2.5	Многопоточность	9
2.6	Графический интерфейс пользователя	9
3	Java Junior	10
3.1	JDBC	10
3.2	Сетевое программирование	10
3.3	Введение в архитектуры приложений на Java	10
3.4	Сериализация, StreamAPI, ReflectionAPI	10
3.5	Введение в Java EE	10
3.6	Введение в Spring framework	10
4	Java Junior+	11

Глава 1

Java Core

1.1. Платформа: история и окружение

Конспект лекции

Краткая история (причины возникновения)

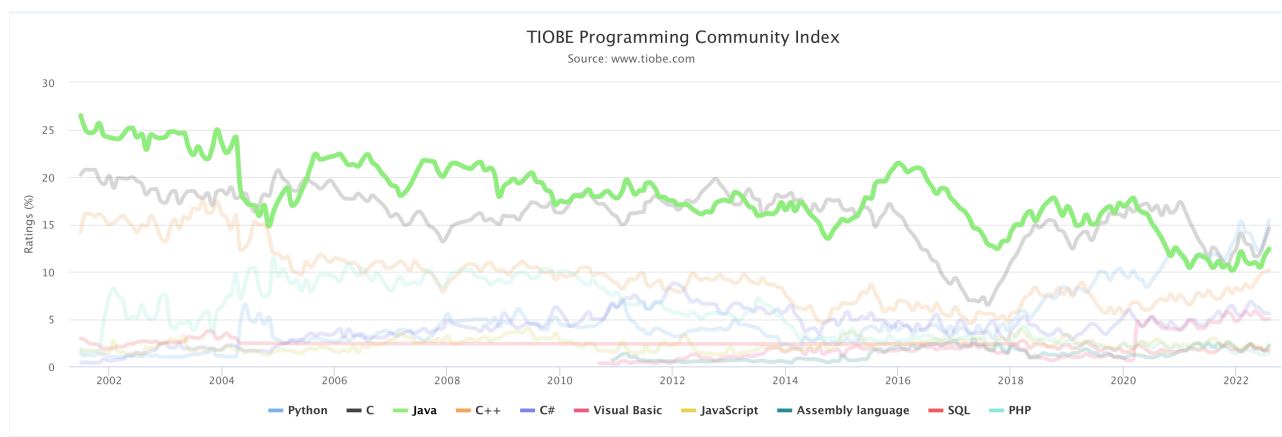


Рис. 1.1: График популярности языков программирования TIOBE

Базовый инструментарий, который понадобится (выбор IDE)

- NetBeans - хороший, добротный инструмент с лёгким ностальгическим оттенком;
- Eclipse - для поклонников Eclipse Foundation и швейцарских ножей с полусотней лезвий;
- IntelliJ IDEA - стандарт де-факто, используется на курсе и в большинстве современных компаний;
- Android Studio - если заниматься мобильной разработкой.

Что нужно скачать, откуда (как выбрать вендора)

Из чего всё состоит (JDK, JRE, JVM, выбор версии)

Структура проекта (пакеты, классы, метод main, комментарии)

Отложим мышки в сторону (CLI: сборка, пакеты, запуск)

Документирование (Javadoc)

Задания к семинару

- Скомпилировать проект из трёх классов, находящихся в двух пакетах, а также создать для этого проекта стандартную веб-страницу с документацией


Сценарий лекции

Платформа: история и окружение

Иван Игоревич Овчинников

GeekBrains. Java Core.

2022



Иван Игоревич Овчинников (GB: Java)Платформа: история и окружение20221 / 4

Добро пожаловать на специализацию Java, лучшую из всех специализаций Java, что вы сможете найти.

Введение и знакомство (о себе)

Иван Овчинников. НПО ИТ, РКС, GB.

Используемые технологии

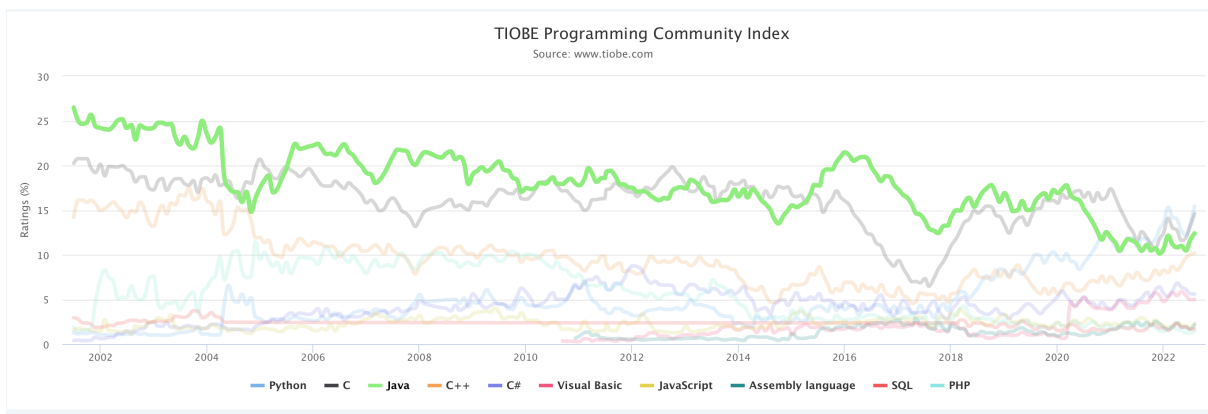
Много и с удовольствием C, C++, Java, Verilog. C# и Python даже не считаем.

Для GB более 20 потоков по разным направлениям, более двух тысяч студентов.



Здравствуйте, меня зовут Иван, я разработчик программного обеспечения в Российских космических системах, начальник группы программистов, автор и преподаватель нескольких курсов в направлении программирования на портале GeekBrains. Пара слов обо мне, почему я нахожусь здесь и планирую Вам что-то рассказывать про язык Java? Для этого придётся вынести за скобки мой опыт работы в цифровой схемотехнике и других языках, в том числе создание бортовой аппаратуры, которая прямо сейчас летает у нас над головами или продолжает доставлять другие полезные грузы в космос. С точки зрения языка Java, обо мне можно сказать, что я являюсь разработчиком нескольких отраслевых информационных систем поддержки единой наземной инфраструктуры российского космического агентства. На моём счету также участие в межотраслевых проектах по созданию единых баз данных космического применения.

Краткая история



Иван Игоревич Овчинников (GB: Java)

Платформа: история и окружение

2022

3 / 4

Итак язык Java. Согласно википедии, Java — строго типизированный объектно-ориентированный язык программирования общего назначения, разработанный компанией Sun Microsystems. Разработка ведётся сообществом; язык и основные реализующие его технологии распространяются по лицензии GPL.

Приложения Java обычно транслируются в специальный байт-код, поэтому они могут работать на любой компьютерной архитектуре, для которой существует реализация виртуальной Java-машины. Дата официального выпуска — 23 мая 1995 года. Традиционно занимает высокие места в рейтингах популярности языков программирования (3-е место в рейтинге ТЮВЕ (на август 2021)). На графике отлично видно, что с 2002 года язык уверенно держится в тройке популярных, значительную часть времени возглавляя его.

Но это всё - сухие факты, ничего не говорящие нам о том, чем руководствовались разработчики языка и что нам, программистам, с этим делать.

Существует, интересная версия происхождения названия языка, связанная с аллюзией на кофе машину как пример бытового устройства, для программирования которого изначально язык создавался. Кстати, в результате работы проекта мир увидел принципиально новое устройство, карманный персональный компьютер, который опередил своё время более чем на 10 лет, но из-за большой стоимости не смог произвести переворот в мире технологии и был забыт. Устройство Star7 не пользовалось популярностью, в отличие от языка программирования Java и его окружения. С середины 1990-х годов язык стал широко использоваться для написания клиентских приложений и серверного программного обеспечения. Тогда же некоторое распространение получила технология Java-апплетов — графических Java-приложений, встраиваемых в веб-страницы; с развитием возможностей динамических веб-страниц технология стала применяться редко и язык стал применяться для бэк-энда.

Итак Язык программирования Java занял свою нишу в системном и прикладном программировании из-за своей высокой скорости работы, и веб-программировании, из-за удобного сетевого окруже-



ния, ставшего основой для создания бэк-энда веб-сервисов. Java очевидно испытал влияние языков C, C++, Pascal и других, так что многое в этих языках достаточно досконально изучить один раз, чтобы иметь возможность понимать все связанные.

Базовый инструментарий

- 1 Eclipse
- 2 NetBeans
- 3 IntelliJ IDEA
- 4 BlueJ
- 5 Oracle JDeveloper
- 6 MyEclipse
- 7 Greenfoot
- 8 jGRASP
- 9 JCreator
- 10 DrJava



Мы привыкли к тому, что для программирования нужна некоторая среда, которая будет удобно подсказывать нам, что делать, как писать программу, запускать написанный код и помогать нам отлаживать его. Чаще всего это так, но я призываю вас не останавливаться на единственном инструменте, ведь если быть всегда сосредоточенным на отвёртке, вы никогда не узнаете, что люди придумали очень удобный шуруповёрт.

Я не планирую рекламировать ту или иную среду, скорее всего не буду даже много говорить о плюсах и минусах той или иной среды, просто покажу список и коротко расскажу о главных действующих лицах:

1. NetBeans - если коротко, то эта среда с нами из уважения к истории, она была первой, созданной для разработки на языке Java. Долгое время разработка этой среды не велась, но в последние несколько лет разработчики как будто бы вспомнили о ней и активно взялись за дело, сейчас есть версии под все популярные ОС, поддерживается весь популярный современный функционал, есть прикольные самобытные моменты, вроде расширенных клавиатурных макросов для написания функций с передаваемыми в них массивами;
2. Eclipse - проще, наверное, назвать язык или технологию, под которую нет плагина для этой весьма расширяемой и дополняемой среды программирования. Долгое время, вплоть до недавнего, а где-то и до сих пор, является корпоративным стандартом для написания Enterprise приложений и прочих сложных корпоративных приложений. Весьма гибкая среда и-за того, что разрабатывается Eclipse Foundation к ней можно подключить плагины, решающие почти лю-

бые задачи разработки. Из того что я видел своими глазами, Siemens использует Eclipse для разработки своих систем поддержки жизненного цикла изделий;

3. IntelliJ IDEA - стандарт де-факто примерно десять последних лет. Не уверен, можно ли назвать эту среду российской разработкой, но у компании, разрабатывающей эту среду, три русских учредителя. Да и сама эта компания началась именно с этой среды. В этой среде можно настроить почти любую деталь, что часто бывает очень удобно. Среда расширяется плагинами на почти все случаи жизни разработчика. Часто входит в экосистему JetBrains, развёрнутую на предприятии, поэтому тесно интегрируется в другие сферы деятельности компании. Существует как бесплатная, так и платная версия с расширенным функционалом, например, доступа к базам данных.

Отдельно стоит обговорить такую среду, как Android Studio, как мог заметить внимательный зритель, её нет в списке на слайде, но не потому, что в ней нельзя разрабатывать Java-приложения, а потому что если не разрабатывать в ней именно Android приложения, следует отказаться от её использования из-за некоторой избыточности. Среда снабжена значительным количеством надстроек и эмуляторов мобильных устройств, которые попросту будут вам мешать, если вы не используете их в своей профессиональной деятельности. С другой стороны, если ваша цель - это именно мобильная разработка под Android, то и выбора у вас особо нет, только Android Studio.

1.2. Управление проектом: сборщики проектов

Управление проектом: Jar-файлы; Gradle/Maven: зависимости, выгрузка артефакта, публичные репозитории, свойства проекта, приватные репозитории (хостинг); Bazel

Задания к семинару

- Создать загружаемый артефакт с функцией суммирования двух чисел и загрузить его в локальный кэш артефактов;

1.3. Специализация: данные и функции

Базовые функции языка: математические операторы, условия, циклы, бинарные операторы; Данные: типы, преобразование типов, константы и переменные (примитивные, ссылочные), бинарное представление, массивы (ссылочная природа массивов, индексация, манипуляция данными); Функции: параметры, возвращаемые значения, перегрузка функций;

Задания к семинару

- Написать как можно больше вариантов функции инвертирования массива единиц и нулей за 15 минут (без ветвлений любого рода);
- Сравнить без условий две даты, представленные в виде трёх чисел гггг-мм-дд;

1.4. Специализация: ООП

Инкапсуляция: Классы и объекты (внутренние классы, вложенные классы, static, private/public, final, интерфейс взаимодействия с объектом), перечисления (создание, конструкторы перечислений, объекты перечислений, дополнительные свойства); Наследование: extends, Object (глобальное наследование), protected, преобразование типов, final; Полиморфизм: override, abstract, final;

1.5. Специализация: Тонкости работы

Файловая система и представление данных; Пакеты `java.io`, `java.nio`, `String`, `StringBuilder`, `string pool`, ?JSON/XML?



Глава 2

Java Development Kit

2.1. Исключения

Механизм и понятие, введение в **многопоточность**, throw; Наследование и полиморфизм в исключениях; Способы обработки исключений (try/catch, throws, finally); try-with-resources, штатные и нештатные ситуации

2.2. Интерфейсы

Понятие и принцип работы, implements; Наследование и множественное наследование интерфейсов; Значения по-умолчанию

2.3. Обобщённое программирование

2.4. Коллекции

List, Set; Хэш-код, интерфейс Comparable; Map, Object (Использование методов, Переопределение методов); Итераторы, **Многопоточные** коллекции

2.5. Многопоточность

Понятие, Принцип (реальная и псевдопараллельность); Runnable, Thread (Свойства, Особенности создания); Deadlock, Состояние гонки, Object (Ожидание/уведомление); Синхронизация (Синхронизация по монитору, Частичная, по классу)

2.6. Графический интерфейс пользователя

GUI (Swing): Окна и свойства окон, **Многопоточность** и абстрагирование асинхронных вызовов; менеджеры размещений и проведение параллелей с веб-фреймворками, разделение на фронт-энд и бэк-энд; JPanel и рисование, Обработка действий пользователя

Глава 3

Java Junior

3.1. JDBC

3.2. Сетевое программирование

Socket, ServerSocket, Многопоточность, абстрагирование сетевого взаимодействия, интерфейсы

3.3. Введение в архитектуры приложений на Java

клиент-серверы, веб-приложения, сервлеты, толстые и тонкие клиенты, выделение бизнес-логики и хранения, создание общих пространств и модульность проекта

3.4. Сериализация, StreamAPI, ReflectionAPI

3.5. Введение в Java EE

3.6. Введение в Spring framework

Глава 4

Java Junior+

