

No description, website, or topics provided.

62 commits

2 branches

0 releases

1 contributor

Branch: master ▾

New pull request

Find file

Clone or download ▾

Tanechka Merge remote-tracking branch 'origin/master'		Latest commit cc7ff4d on 23 Oct 2017
config	Lesson17 DB_credentials_as_resources	9 months ago
src/ru/javawebinar/basejava	Lesson17 DB_credentials_as_resources	9 months ago
test/ru/javawebinar/basejava	Lesson17 HW16	9 months ago
web	Lesson17 HW16_add_resume	9 months ago
.gitignore	Lesson1_initial	a year ago
README.md	Update README.md	9 months ago

README.md

Курс BaseJava (обновленный и переработанный)

Разработка Web приложения "База данных резюме"

- используем: Java 8, [IntelliJ IDEA](#), [GitHib/Git](#), Сервлеты, JSP, JSTL, Tomcat, JUnit, PostgreSQL, GSON, JAXB
- хранение резюме
 - в памяти на основе массива, отсортированного массива, списка и ассоциированного массива (Map)
 - в файловой системе (File API и [Java 7 NIO File API](#))
 - в стандартной и кастомной сериализации Java
 - в формате JSON ([Google Gson](#))
 - в формате XML ([JAXB](#))
 - в реляционной базе [PostgreSQL](#)
- деплой веб приложения
 - в контейнер сервлетов [Tomcat](#)
 - в облачный сервис [Heroku](#)

Любое знание стоит воспринимать как подобие семантического дерева: убедитесь в том, что понимаете фундаментальные принципы, то есть ствол и крупные ветки, прежде чем лезть в мелкие листья-детали. Иначе последним не на чем будет держаться.

*— Илон Маск

Программа

Занятие 1

- Презентация проекта
- Разработка ПО
- Обзор языка Java
- Системы управления версиями. Git
- ПЕРВОЕ ДОМАШНЕЕ ЗАДАНИЕ

Занятие 2

- Принципы ООП
- Структура памяти: куча, стек, регистры, константы

- Типы данных. Пакеты

Занятие 3

- Объектная модель в Java
- Сложность алгоритмов
- Паттерн проектирования Шаблонный метод

Занятие 4

- Работа со строками
- Исключения
- Reflection. Аннотации. Модульное тестирование

Занятие 5

- Контейнеры/коллекции

Занятие 6

- Iterator / Iterable. Вложенные, внутренние, локальные и анонимные классы
- Новое в Java 8

Занятие 7

- Параметризация. Стирание типов
- Логирование
- Синглтон, Enum

Занятие 8

- Работа с датами и временем
- Работа с файлами и ресурсами

Занятие 9

- Ввод/вывод
- Сериализация
- NIO
- Основы Java 8 Stream API

Занятие 10

- Формат XML. Работа с XML в Java
- JSON
- DataInputStream / DataOutputStream

Занятие 11

- Многопоточность. Параллельное выполнение.
- Потоки. Синхронизация
- Ленивая инициализация, JMM

Занятие 12

- java.util.concurrent

Занятие 13

- Базы данных. Реляционные СУБД. PostgreSQL
- Конфигурирование данных в Java проекте
- Подключение DB в проект

Занятие 14

- JOIN
- Транзакции
- Установка/запуск Tomcat

Занятие 15

- HTML, Tomcat
- Сервлеты

Занятие 16

- JSP
- JSTL

Занятие 17

- Деплой в Heroku
- Classloader
- Обзор Java Enterprise

Рекомендуемые книги

- [YAKOV FAIN: Программирование на Java для начинающих](#)
- [Книги по Java: от новичка до профессионала](#)
- [Джошуа Блох: Java. Эффективное программирование, 2-е издание](#)
- [Гамма, Хелм, Джонсон: Приемы объектно-ориентированного проектирования. Паттерны проектирования](#)
- [Редмонд Э.: Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL.](#)

Ресурсы в сети

- [Руководство по Java Core](#)
- [Java. Базовый курс](#)
- [intuit: Программирование на Java](#)
- [Основы программирования на Java: учебное пособие](#)

ichimax Update lesson01.md

ed45d60 20 days ago

2 contributors 

115 lines (93 sloc) | 11.8 KB

Первое занятие

Вебинар-презентация обучения на проекте BaseJava

Итоговый пример приложения, разрабатываемого в рамках данного курса (на примере резюме Григория Кислина)

Подготовка рабочего окружения

- Установите [JDK8](#) (выбрать Accept License Agreement)
- Установите систему управления версиями [Git](#) (опции по умолчанию)
- Создайте аккаунт на [GitHub](#)
- Для удобной навигации по файлам на GitHub установите расширение для браузера — [Octotree](#)
- Установите [IntelliJ IDEA](#) (пока нет базы данных и веб, можно работать с версией Community)

Версия Ultimate является платным продуктом, рассчитанным на 30 дней пробного бесплатного использования. Для выполнения заданий в этой IDE каждый участник курса получает единовременный ключ на 6 месяцев

1. Разработка ПО

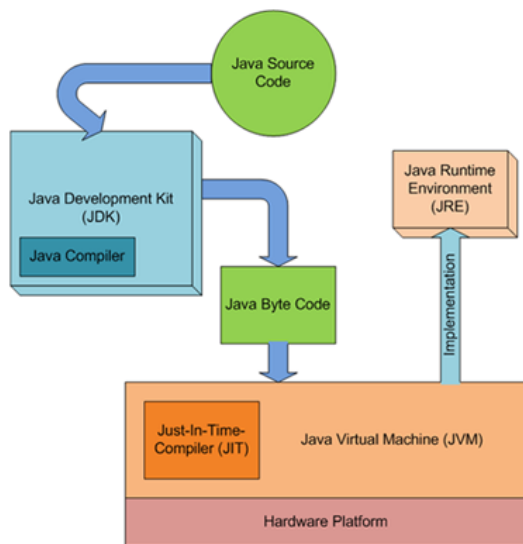
- Мифический человеко-месяц (wiki)
- Размеры проектов в количестве строк кода [eng]
- Соглашения по оформлению кода
- Методологии разработки ПО
- Ещё раз про семь основных методологий разработки

2. Обзор инструментов и технологий

- Bash — шпаргалка для начинающих
- Интерактивный курс по SQL
- Что и почему используют Java-разработчики: опрос RebelLabs
- Java in 2017 Survey Results
- Дополнительно:
 - Автоматизированная сборка проекта в Java

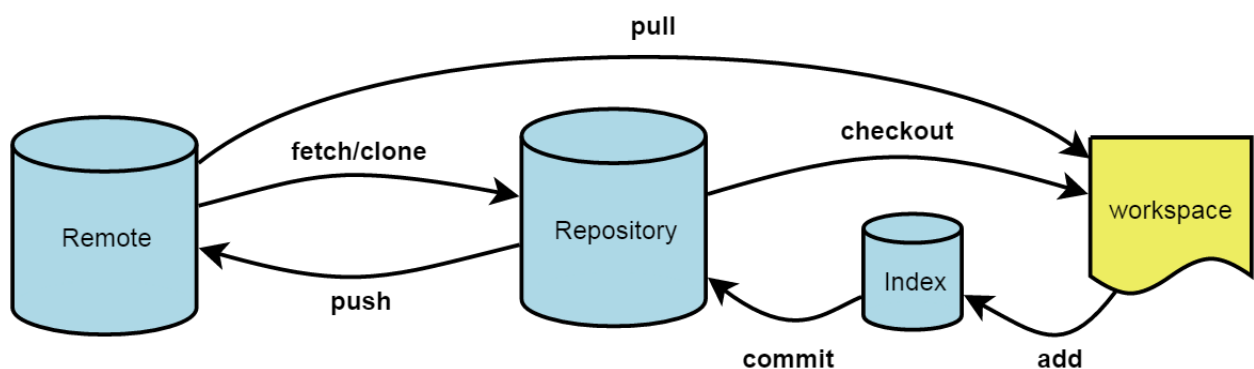
3. Обзор языка Java

- Java, JVM, JIT-компиляция (wiki)
- Что такое Java? История создания
- Programming languages TIOBE Index



- [ME, SE, EE](#) (wiki)
- [Java Microbenchmark JMH](#) (используем на курсе [MasterJava](#))
- [Oracle Java8 Home](#)
- Дополнительно:
 - [Java version and vendor data analyzed 2017](#)
 - [Most popular Java application servers: 2017 edition](#)

4. Системы управления версиями. Git



- [Система управления версиями](#) (wiki)
- [Сравнение разных VCS](#)
- [Git обучалка](#)
- [Интерактивная Git обучалка](#) (в настройках выберите русский язык)
- [Официальная книга про Git](#)
- [Working with remote repositories](#)
- [Базовый курс по обучению Git](#) (youtube)
- [Git. Быстрый старт](#) (youtube)

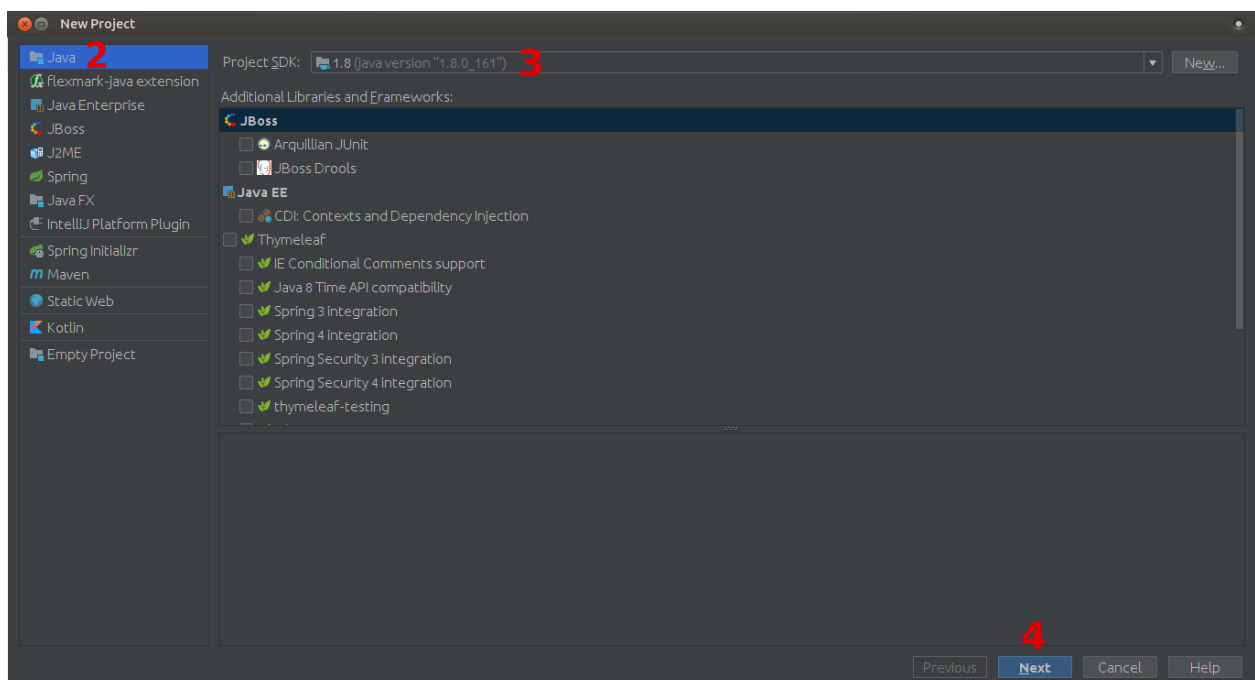
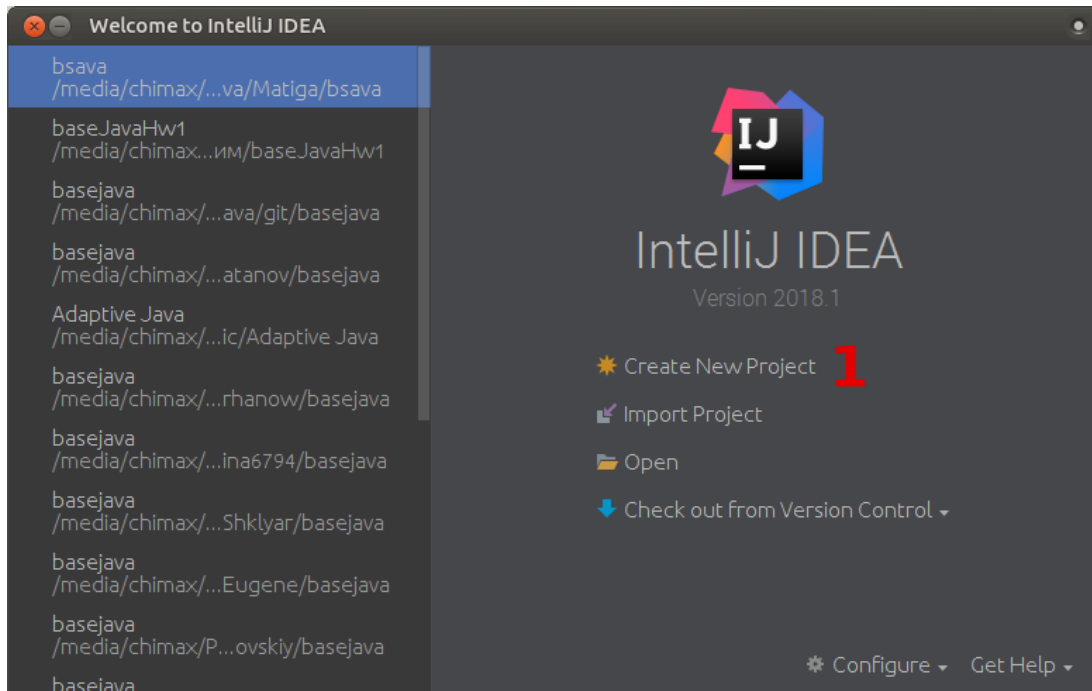
Настройка проекта

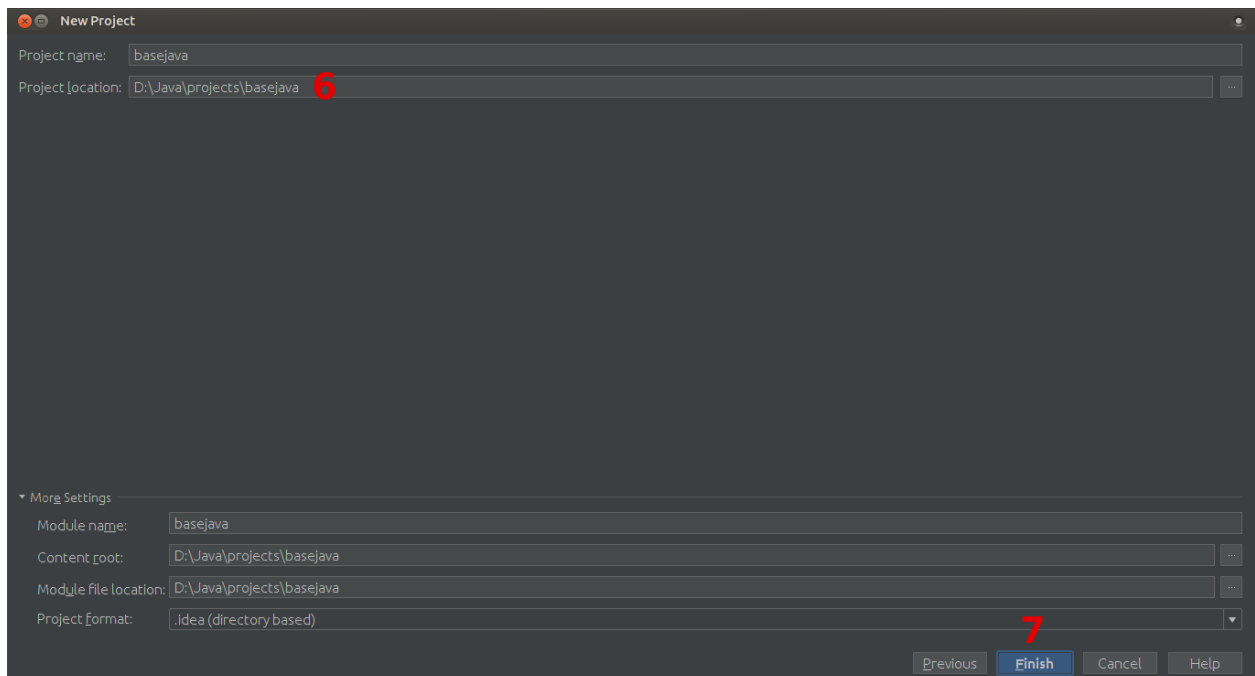
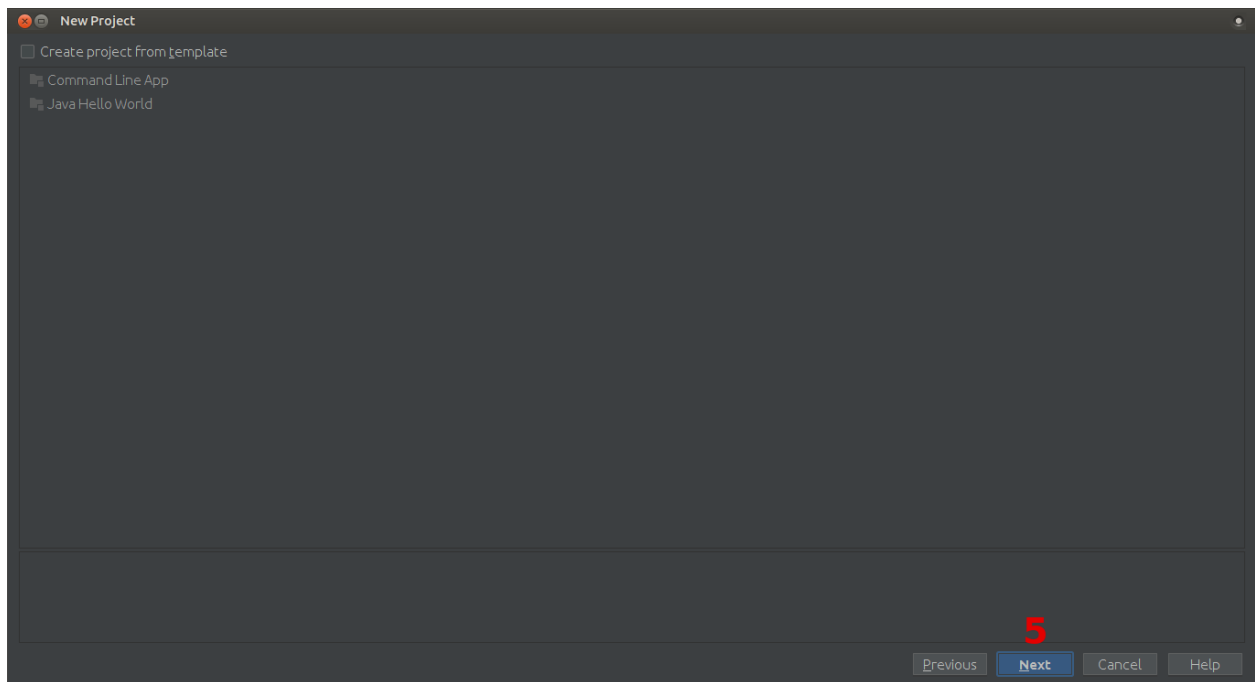
- Создайте на GitHub репозиторий с названием `basejava`
- Откройте консоль (терминал) у себя на компьютере
- Если необходимо, [занесите](#) git в переменную окружения PATH и перезапустите консоль
- Создайте локальную копию проекта: `git clone https://github.com/JavaOps/basejava.git`
- Перейдите в каталог проекта: `cd basejava`
- Настройте git в локальном проекте на свой проект в GitHub:
 - `git remote -v`
 - `git remote set-url origin url_на_твой_basejava-репозиторий.git` - настройка pull
 - `git remote set-url --push origin url_на_твой_basejava-репозиторий.git` - настройка push

o `git push -u origin master`

Домашнее задание HW1

- Создайте в IntelliJ IDEA новый проект, выбрав каталог `basejava`, который вы клонировали ранее к себе на компьютер:





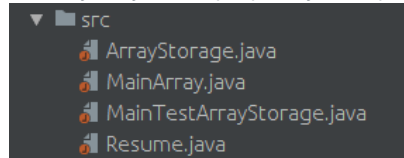
- Реализуйте класс `ArrayStorage`, организовав хранение резюме на основе массива с методами `save`, `get`, `delete`, `size`, `clear`, `getAll`
- При этом храните все резюме в начале `storage` (без дырок в виде `null`), чтобы не перебирать каждый раз все 10000 элементов

Схема хранения резюме в `storage` (от 0 до `size - 1` элементов `null` нет):
`r1, r2, r3, ..., rn, null, null, ..., null`
<----- `size` ----->
<----- `storage.length (10000)` ----->

- Посмотрите на класс `java.util.Arrays`. В нем есть полезные методы, которые помогут вам написать более простой и понятный код
- Протестируйте вашу реализацию с помощью классов `MainArray.main()` и `MainTestArrayStorage.main()`
- Изучите дополнительные материалы по IntelliJ IDEA:
 - [Idea Wiki](#) (поставьте кодировку UTF-8, поменяйте шрифт по умолчанию на DejaVu)
 - [Руководство пользователя IntelliJ IDEA. Отладчик](#)
 - [Эффективная работа с кодом в IntelliJ IDEA](#) (youtube)
 - [Эффективная работа в IDEA](#) (youtube)

Вопросы по HW1

Не могу запустить программу, да и рядом с классами появился какой-то значок



- Проблема в том, что IDEA неправильно "воспринимает" папку `src`. Для ее решения необходимо нажать ПКМ на папке `src` -> выбрать `Mark Directory as` -> `Sources Root`

Что такое null?

- [Что такое null в Java?](#) (оригинал: [What is null in Java?](#))

Замечания по выполнению HW1

1. При реализации метода `delete()` сортировать резюме по `uuid` не требуется
2. Все резюме в хранилище имеют уникальный `uuid`, что исключает повторы
3. Давайте осмысленные комментарии коммитам
4. Перед каждым коммитом не забывайте пользоваться сочетанием клавиш `Ctrl + Alt + L` (автоматическое форматирование кода)
5. Не злоупотребляйте пустыми строками — выработай какой-то единый стиль их написания
6. Удаляйте неиспользуемые импорты (`Ctrl + Alt + O`)
7. Не игнорируйте подсказки IDEA (подсвечивает)
8. `Resume r` — давайте переменным осмысленные имена, например `resume . r` допустимо в коротких циклах и лямбдах
9. В методе `clear()` обнуление массива предполагает именно обнуление (null), а не создание нового
10. По заданию необходимо посмотреть в классе `Arrays` методы, которые помогут упростить реализацию ваших методов `clear()` и `getAll()`

 ichimax Update lesson02.md

959a773 7 days ago

2 contributors  

58 lines (51 sloc) | 6.18 KB

Второе занятие

разбор HW1 будет на следующем уроке вместе с HW2

1. Принципы ООП

- [Объектно-ориентированное программирование \(wiki\)](#)
- [Объектно-ориентированное программирование \(перевод статьи\)](#)
- [Основы Объектно-Ориентированного Программирования \(ООП\)](#)
- [Наследование, агрегация, композиция, ассоциация \(wiki\)](#)
- [Типы отношений между классами](#)
- [Достоинства / Недостатки ООП](#)
- **Дополнительно:**
 - [Зачем нам ООП и что это такое?](#)
 - [Николай Алименков — Парадигмы ООП \(youtube\)](#)
 - [Object-Oriented Programming Concepts](#)
 - [Classes and Objects](#)

2. Структура памяти: куча, стек, регистры, константы

- [Стек \(wiki\)](#)
- [От Java-кода к Java-куче](#)
- [Java Heap Space vs Stack – Memory Allocation in Java](#)
- [Понимаем основы Java garbage collection](#)
- **Дополнительно:**
 - [Из каких частей состоит память java процесса](#)
 - [Permanent область памяти](#)
 - [Java thread stack](#)
 - [Размер Java объектов](#)
 - [JVM - краткий курс общей анатомии \(youtube\)](#)
 - [What and where are the stack and heap?](#)
 - [The Java Virtual Machine Specification Java SE 8 Edition](#)

Начиная с Java 8 область памяти `PermGen` была упразднена и заменена на `Metaspace`

3. Типы данных. Пакеты

- [Типы данных](#)
- [Классы-обертки](#)
- [Java types \(youtube\)](#)
- [Модификаторы доступа \(youtube\)](#)
- [Пакеты \(youtube\)](#)
- **Дополнительно:**
 - [Packages](#)

- [Primitive data types](#)
- [What is a NullPointerException, and how do I fix it?](#)
- [Why should one use Objects.requireNonNull\(\)?](#)
- [Инициализация и загрузка классов](#) (youtube)

Домашнее задание HW2

Правка к видео: в `ArrayStorage.delete()` ВМЕСТО `storage[i] = null` нужно записать `storage[size - 1] = null`

- Еще раз прочитайте, [как правильно оформлять код](#)
- [Настройте IDEA](#) на автоматическое удаление неиспользуемых импортов
- Реализуйте и протестируйте `ArrayStorage.update(Resume resume)`
- Сделайте проверки:
 - в `update/delete/get` - резюме есть в `storage` ?
 - в `save` - резюме нет в `storage` ?
 - сделайте в `save` проверку на переполнение
 - выведите соответствующие предупреждения для всех, указанных выше проверок - `System.out.println("Resume ...")`
- Избавьтесь от дублирования в коде `ArrayStorage`

 ichimax Update lesson03.md

78fb092 7 days ago

2 contributors 

60 lines (47 sloc) | 5.79 KB

Третье занятие

Разбор домашнего задания

Коммиты:

- `HW1`
- `Introduce package and encapsulation`
- `lesson03 HW02`

1. Объектная модель в Java

Коммиты:

- `Lesson03 equals&hashCode, static`
- `Lesson03 implements interface`

Ресурсы:

- Модификатор `static`
- 10 заметок о модификаторе `static` в Java
- Класс `Object`. Контракт `equals/hashCode`
- Абстрактные классы (youtube)
- Интерфейсы
- Полиморфизм
- Отличия абстрактного класса от интерфейса
- Дополнительно:
 - Что такое полиморфизм?
 - Ключевое слово `static` (youtube)
 - Интерфейсы, абстрактные классы, полиморфизм (youtube)
 - Разбираемся с `hashCode()` и `equals()`

2. Сложность алгоритмов

- Временная сложность алгоритма (wiki)
- Вычислительная сложность (wiki)
- Алгоритмы и структуры данных для начинающих: сложность алгоритмов
- Time complexity
- Бинарное дерево поиска (youtube)
- Дополнительно
 - Легко написанная книга по алгоритмам и их сложности [Грокаем алгоритмы](#)

3. Паттерн проектирования Шаблонный метод

Коммиты:

- [Lesson03 abstract storage & sorted storage](#)

Ресурсы:

- [Паттерн проектирования — Шаблонный метод](#) (wiki)
- [Шаблонный метод \(Template Method\)](#) (youtube)
- [Шаблонный метод](#)
- **Дополнительно**
 - Книга из известной серии Head First — [Паттерны проектирования](#)



Домашнее задание HW3

- Перед тем, как приступить к ДЗ, приведите свой проект в соответствии с коммитами. Обратите внимание, что `packages` в видео и коммитах — отличаются
- Закончите реализацию `AbstractArrayStorage`, `ArrayStorage`, `SortedArrayStorage`, используя паттерн Шаблонный метод
- В `SortedArrayStorage` храните элементы отсортированными:
 - сортировать весь массив не надо
 - не используйте `Arrays.sort()`
 - обратите внимание на бинарный поиск

 ichimax Update lesson04.md

e231fa9 24 days ago

2 contributors 

66 lines (54 sloc) | 5.97 KB

Четвертое занятие

Разбор домашнего задания

Коммиты:

- [lesson04 HW03](#)

1. Работа со строками

Коммиты:

- [Lesson04 String](#)

Ресурсы:

- [Строки в Java](#)
- [Кодировка в Java](#)
- [Ошибки при использовании строк](#)
- [Обработка строк в Java](#)
- Дополнительно:
 - [StringBuilder vs StringBuffer](#)
 - [String vs StringBuffer vs StringBuilder](#)
 - [String literal pool](#)

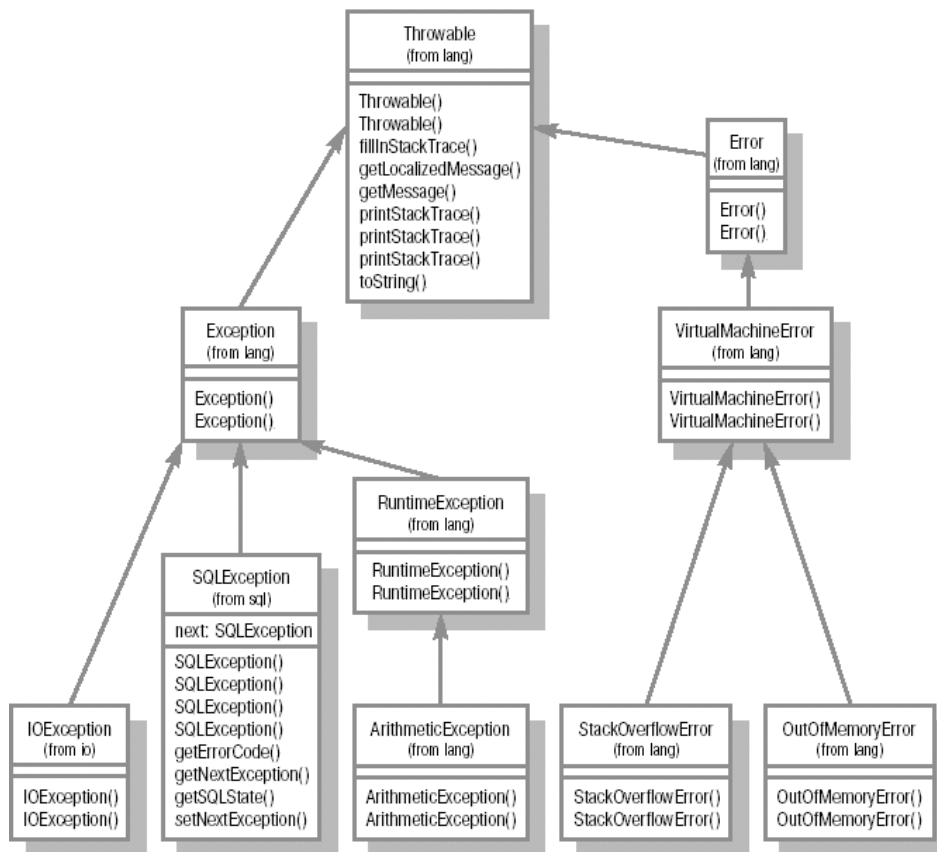
2. Исключения

Коммиты:

- [Lesson04 Exceptions](#)

Ресурсы:

- [Исключения \(Exceptions\)](#)
- [Статья про исключения](#)
- Про исключения также можно почитать в книге Джошуа Блоха - "[Java. Эффективное программирование](#)"
- [Конструктор](#)
- [Ключевые слова: this, super](#)
- Дополнительно:
 - [Exceptions](#)
 - [Checked vs unchecked exception explanation](#)



3. Reflection. Аннотации. Модульное тестирование

Коммиты:

- [Lesson04 Reflection and JUnit](#)

Ресурсы:

- [Java 8 аннотации](#)
- [Руководство по Java Reflection API](#)
- [Java Reflection Example Tutorial](#)
- Дополнительно:
 - [The Reflection API](#)
 - [What is reflection and why is it useful?](#)

В нашем проекте Reflection используют JUnit и будут использовать библиотеки работы с XML и JSON

- [Фреймворк для модульного тестирования JUnit](#)
- [Тестирование с помощью JUnit \(Test Case\)](#)
- [Тестирование кода Java с помощью фреймворка JUnit \(youtube\)](#)

Домашнее задание HW4

- Реализуйте тесты: `AbstractStorageTest`, `ArrayStorageTest` и `SortedArrayStorageTest`
- В `MainReflection` сделайте вызов метода `toString` через отражение, и выведите результат

Подсказки по HW4

- `SortedArrayStorageTest` должен запускаться с `SortedArrayStorage`
- `ArrayStorageTest` с `ArrayStorage`
- тестировать правильность сортировки не надо
- логика реализации теста на переполнение массива (`StorageException`):
 - если исключение вылетит раньше наполнения, то тест должен провалиться (см. `Assert.fail()`)

- если исключение вылетает, когда пытаемся добавить еще одно резюме в полностью заполненный массив - тест пройден
- добавьте конструктор в `AbstractArrayStorageTest`, который инициализирует `Storage storage`, а в наследниках добавьте конструкторы, которые будут вызывать `super()` с нужным хранилищем

 ichimax Update lesson05.md

6a71343 23 days ago

2 contributors 

34 lines (25 sloc) | 3.45 KB

Пятое занятие

Разбор домашнего задания

Коммиты:

- `lesson05 HW04`

В коммите содержится ошибка: в методе `AbstractArrayStorageTest.saveOverflow` цикл идет до `STORAGE_LIMIT + 1` включительно, а надо до `STORAGE_LIMIT`

Контейнеры/коллекции

Коммиты:

- `Lesson05 Static initializer block and Collections`

Ресурсы:

- Контейнеры/коллекции
- [List, Set, Map, Queue, Iterator, ListIterator](#)
- Структуры данных в картинках
- Инициализация полей в Java
- Java собеседование по коллекциям
- Часто задаваемые на собеседованиях вопросы по классам коллекциям в Java
- Собеседование по Java — коллекции
- `Collection.toArray(new T[0])` or `Collection.toArray(new T[size])`, that's the question

Домашнее задание

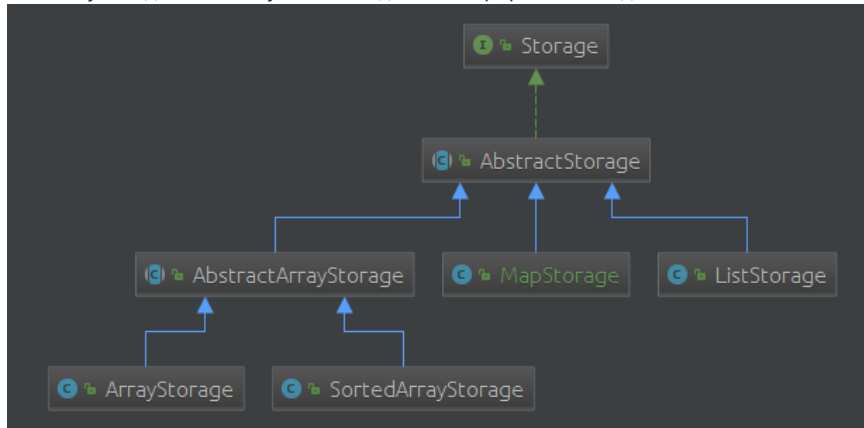
- Выделите общий класс `AbstractStorage` и реализуйте подклассы `ListStorage` и `MapStorage`. Для этого вам необходимо вынести в `AbstractStorage` максимум кода, исключив тем самым его дублирование (полностью реализацию `MapStorage` рассмотрим на след уроке, в данном ДЗ достаточно сделать заглушки для методов)
- сделайте класс для тестирования `ListStorage`

Выбор реализации `List` и `Map` за вами

- [Шаблонный метод \(шаблон проектирования\)](#)

Поведенческий шаблон проектирования, определяющий основу алгоритма и позволяющий наследникам переопределять некоторые шаги алгоритма, не изменяя его структуру в целом

В итоге у вас должна получиться подобная иерархия наследования классов



 ichimax Update lesson06.md

725ea32 16 days ago

1 contributor

58 lines (48 sloc) | 4.81 KB

Шестое занятие

Разбор домашнего задания

Коммиты:

- `lesson06 HW05`

Ресурсы:

- Автоупаковка и распаковка в Java
- Преобразования Integer и int
- Класс-обертка (youtube)
- Дополнительно:
 - Autoboxing and Unboxing
 - Why do we use autoboxing and unboxing in Java?

1. Iterator / Iterable. Вложенные, внутренние, локальные и анонимные классы

Коммиты:

- `Lesson06 Iterator, Comparator and Inner Classes`

Ресурсы:

- Паттерн проектирования Итератор
- Iterator / Iterable
- Дополнительно:
 - Iterator in Java
- Вложенные классы в Java
- Вложенные, внутренние, локальные и анонимные классы
- Локальные и анонимные классы.
- Интерфейсы Comparable и Comparator
- Анонимные классы в Java (youtube)
- Nested (static) классы в Java (youtube)
- Inner (non-static) классы в Java (youtube)
- Дополнительно
 - Вложенные, внутренние, локальные и анонимные классы [eng]

2. Новое в Java 8

Коммиты:

- `Lesson06 Lambda`

Ресурсы:

- [Нововведения в java 8](#)
- [Лямбда-выражения в Java 8 \(youtube\)](#)
- [Функциональный интерфейс](#)
- [Лямбды в Java 8](#)
- Дополнительно:
 - [Java 8 Functional Interfaces](#)
 - [Lambda expressions](#)



Домашнее задание HW6

- Рефакторинг:
 - метод `saveOverflow()` должен исполняться только для массивов
 - в конструктор `Resume` добавьте второй параметр — `fullName`
 - во всех реализациях `Storage` замените метод `Resume[] getAll()` на `List<Resume> getAllSorted()`
- Реализуйте до конца `MapUuidStorage`
- Подумайте, что еще, может выступать в качестве `search key` в реализации на основе `Map`. Выразите свои идеи в коде
- Переделайте компаратор, учтя тот случай, что `fullName` разных людей может совпадать
- Из-за того, что количество тестовых классов растет, воспользуйтесь [аннотациями JUnit](#), которые помогут упростить их запуск

 ichimax Update lesson07.md

b4f0702 on 31 Jan

2 contributors 

67 lines (54 sloc) | 5.86 KB

Седьмое занятие

Разбор домашнего задания

Коммиты:

- Lesson07_HW06 test refactoring
- Lesson07_HW06 add fullName
- Lesson07_HW06 getAllSorted
- Lesson07_HW06 mapStorage

1. Параметризация. Стирание типов

Коммиты:

- Lesson07 generic

Ресурсы:

- Дженерики (Java, обучающая статья)
- Обобщения (Generic)
- Ограничения
- Дополнительно
 - Java Generics Example Tutorial

2. Логирование

Коммиты:

- Lesson07 logging

Ресурсы:

- Log4J (Apache logging)
- Java Logging API - Tutorial
- Логирование в Java / quick start
- Ведение лога приложения
- Java Logging: история кошмара

3. Синглтон. Enum

Коммиты:

- Lesson07 enum

Ресурсы:

- Одиночка (шаблон проектирования)
- Перечисляемые типы (enum) в Java



Домашнее задание:

Доменный объект

- Сделать объектную модель резюме (диаграмма и классы). [Образец резюме](#)
 - Делать только классы, включаемые в Resume. Resume - главный класс. В него все включается (композиция - строгий вид агрегации)
 - Схожие по структуре и функциональности сущности делаем одним классом
 - Модель максимально упрощаем, и храним в ней только необходимые данные, а также функционал для вывода и редактирования резюме
 - В модели резюме должны быть представлены контакты и следующие секции:
 - PERSONAL("Личные качества")
 - OBJECTIVE("Позиция")
 - ACHIEVEMENT("Достижения")
 - QUALIFICATIONS("Квалификация")
 - EXPERIENCE("Опыт работы")
 - EDUCATION("Образование")
 - В секциях Достижения и Квалификация хранить список строк
 - Учесть в классах модели, что обработка резюме (вывод в html, сохранение, чтение) будет происходить следующим образом: обработка `fullName`, цикл обработки по контактам, цикл обработки по секциям (для секций использовать полиморфизм, как для фигур: круг, квадрат...)
 - При добавлении / удалении новых видов контактов (например домашний телефон) или разделов изменения в коде (и БД) должны быть минимальны
- ПРОВЕРЬТЕ свою модель - сделайте объект `Resume` со всеми разделами

Инструменты для рисования:

- [Generate class diagram in IntelliJ IDEA \(Help: working with Diagrams\)](#)
- Нарисовать и сфотографировать
- Online: www.draw.io
- [yEd - Graph Editor](#)

Восьмое занятие

Разбор Домашнего Задания-7

см. коммит Lesson08 HW07 model implementation

- [Объектная модель резюме](#)

1. Работа с датами и временем.

см. коммит Lesson08 DateTime api

- Класс Date, Calendar, TimeZone
- Java – Convert date and time between timezone
- Joda Time library
- Java 8 Date API

2. Работа с файлами и ресурсами.

см. коммит Lesson08 File and Resources

- [File](#). Работа с файловой системой.
- Работа с ресурсами. [Правильно освобождаем ресурсы в Java](#)
- [Java 7 try-with-resources](#)

Домашнее задание:

см. коммит Lesson08 HW08

1. Переделать модель резюме: учесть, что на одной работе (в одном учебном заведении) можно работать/ учиться в разные периоды и при этом имя организации не дублируется
2. Сделать рекурсивный обход и вывод имени файлов в каталогах и подкаталогах (корневой каталог- ваш проект)
3. Заполнить в `AbstractStorageTest` резюме данными
4. Реализовать `AbstractFileStorage` , базовый класс для хранения резюме в файлах.

 ichimax Update lesson09.md

37f46cf on 30 Apr

2 contributors  

34 lines (22 sloc) | 3.28 KB

Девятое занятие

Разбор Домашнего Задания-8

[см. коммит Lesson09 HW08](#)

1. Ввод/вывод.

- Пакет `java.io`
- Паттерн Декоратор.
- Классы `Reader` и `Writer`.

2. Сериализация.

[см. коммит Lesson09 ObjectOutputStream](#)

- Сериализация объектов (serialization)
- Реализация `Storage` используя сериализацию.
- Сериализация: [1](#), [2](#) (youtube)

3. NIO.

[см. коммит Lesson09 AbstractPathStorage](#)

- NIO Java 7
- Чтения строк из файла

4. Основы Java 8 Stream API.

- Потоки

Домашнее задание

- Сделать рекурсивный вывод каталогов с отступом
- Реализовать `ObjectStreamPathStorage` (через `java.nio.file.Path`) и добавить `ObjectStreamPathStorageTest`
- Сделать реализации `Storage` сохранения в файл через `File` и `Path` с возможностью выбора стратегии сериализации (посмотрите на [паттерн стратегия](#)). Кроме сохранения через `ObjectOutputStream/ObjectInputStream` у нас будут еще несколько вариантов сериализации. Сделать тесты для тестирования сохранения через `ObjectOutputStream/ObjectInputStream` для `File` и `Path`.

 ichimax Update lesson10.md

c723f25 on 31 Jan

2 contributors 

37 lines (27 sloc) | 2.76 KB

Десятое занятие

Разбор домашнего задания

Коммиты:

- Lesson10 HW09

Ресурсы:

- Паттерн - Стратегия

1. Формат XML. Работа с XML в Java

Коммиты:

- Lesson10 xml

Ресурсы:

- XML формат и технологии
- Wiki: XML, XSL, DOM, SAX, StAX, JAXB
- Работа с XML в Java. Реализация хранения в XML.

2. JSON

Коммиты:

- Lesson10 json

Ресурсы:

- JSON. JSON в JavaScript
- Работа с JSON в Java: Google GSON и Jackson. Реализация хранения в JSON
- Введение в JSON (youtube)

3. DataInputStream / DataOutputStream

Коммиты:

- Lesson10 DataStream

Ресурсы:

- Реализация Storage на основе DataInputStream/DataOutputStream .

Домашнее задание

- Сделать и протестировать реализацию DataStreamSerializer

ichimax Update lesson11.md

61c55f5 on 4 Feb

2 contributors 

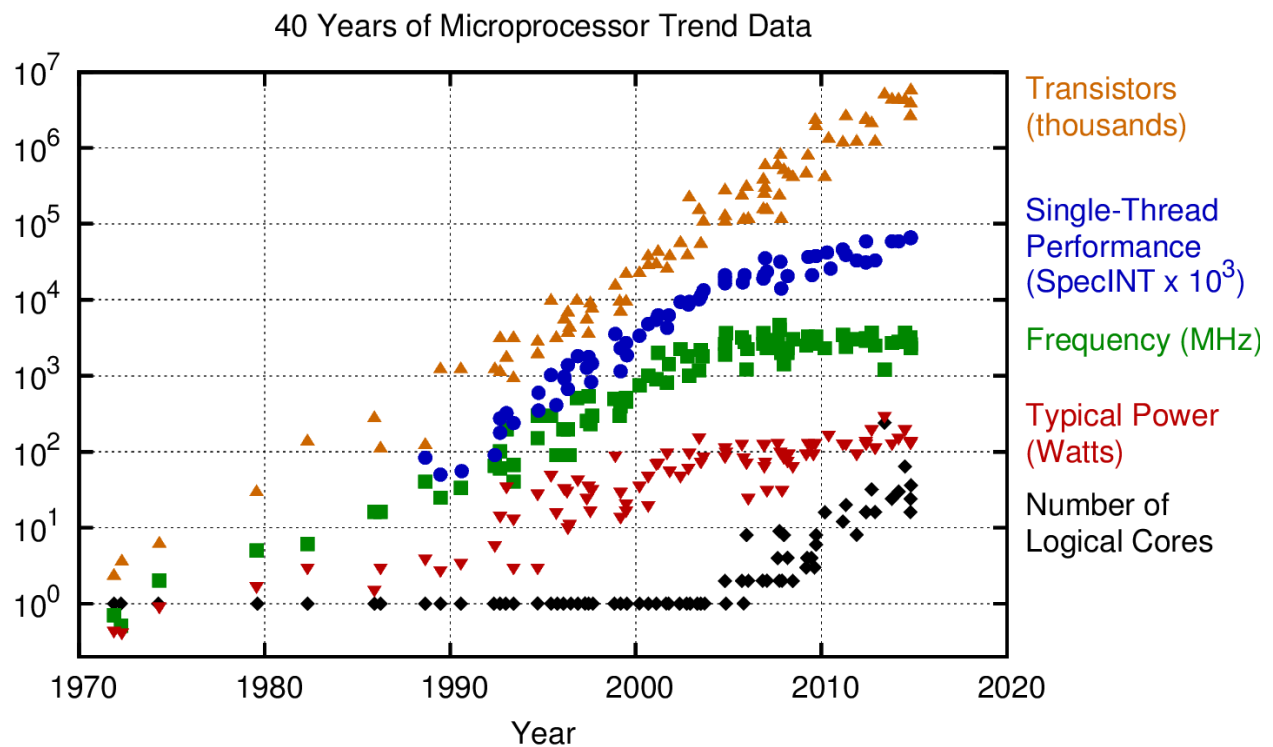
38 lines (28 sloc) | 3.33 KB

Одиннадцатое занятие

Разбор Домашнего Задания-10 будет на следующем уроке

1. Многопоточность. Параллельное выполнение.

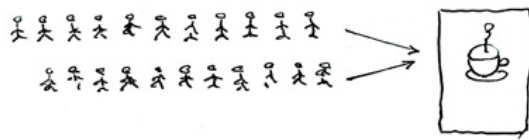
[см. коммит Lesson11](#)



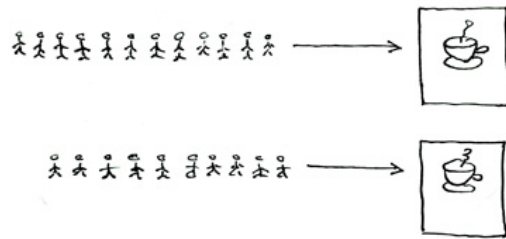
Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

- [Закон Мура](#)
- [Закон Амдала](#)
- [Фундаментальный поворот к параллелизму в программировании](#)

Concurrent = Two Queues One Coffee Machine



Parallel = Two Queues Two Coffee Machines



© Joe Armstrong 2013

Потоки. Синхронизация

- [Потоки выполнения. Синхронизация.](#)
- Методы `wait()`, `notify()`, `notifyAll()` класса `Object`



2. Ленивая инициализация, JMM.

см. коммит [Lesson11](#)

- [Параллелизм в Java](#)
- [Реализация Singleton в JAVA](#)
- [Double checked locking](#)
- [Java Memory Model](#). `final`, `volatile`
- [Initialization-on-demand holder idiom](#)

Ресурсы

- Алексей Владыкин, [Основы многопоточность в Java](#)
- Виталий Чибриков, [Java. Многопоточность](#)
- Computer Science Center, курс [Параллельное программирование](#)
- Юрий Ткач, курс [Advanced Java - Concurrency](#)
- Головач, курс [Java Multithreading](#)

Домашнее задание

- Реализовать deadlock
- [Взаимная блокировка](#)

Двенадцатое занятие

Разбор Домашнего Задания-11

[см. коммит Lesson12 HW11](#)

Concurrency

[см. коммит Lesson12 Concurrency](#)

- [Обзор java.util.concurrent.*](#)
- [Compare-and-swap](#)
- [Справочник по синхронизаторам java.util.concurrent.*](#)
- [Использование ThreadLocal переменных](#)

Замечания по видео:

```
ThreadLocal<SimpleDateFormat> DATE_FORMAT = new ThreadLocal<SimpleDateFormat>() {  
    @Override  
    protected SimpleDateFormat initialValue() {  
        return new SimpleDateFormat("dd.MM.yyyy HH:mm:ss");  
    };  
};
```

можно написать через лямбду:

```
ThreadLocal.withInitial(() -> new SimpleDateFormat("dd.MM.yyyy HH:mm:ss"));
```

А лучше использовать потокобезопасный `DateTimeFormatter` Java 8 Time API:

```
DateTimeFormatter.ofPattern("dd.MM.yyyy HH:mm:ss");
```

Разбор Домашнего Задания-10

[см. коммит Lesson12 HW10](#)

Домашнее задание:

- [Установить PostgreSQL](#)
- Посмотреть на реляционные базы данных и SQL:
 - [Введение в базы данных](#)
 - [Основы SQL](#)

 ichimax Update lesson13.md

57d85cb on 7 Apr

2 contributors 

49 lines (36 sloc) | 3.77 KB

Тринадцатое занятие



1. Базы данных. Реляционные СУБД. PostgreSQL

Коммиты:

- Lesson13 Init DB

Ресурсы:

- DB-Engines Ranking
- Реляционная СУБД (wiki)
- Введение в базы данных
- Реляционные базы vs NoSQL. SQL. Денормализация. PK, FK, Cascade
- PostgreSQL: надёжность
- Работа с базами данных из IDEA
- IDEA Database tools
- Как работает реляционная БД
- SQL ключи во всех подробностях
- Книги по postgresQL
- Интерактивная обучалка по postgresQL



2. Конфигурирование данных в Java проекте

Коммиты:

- Lesson13 properties

Ресурсы:

- Properties sample
- Конфигурирование DB и каталога хранения



3. Подключение DB в проект

Коммиты:

- Lesson13 SqlStorage

Ресурсы:

- JDBC
- JDBC Architecture
- Книга: Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL
- Работа с базами данных с помощью JDBC драйвера
- Уроки по JDBC



Домашнее задание HW13

- Доделать `SqlStorage` без контактов и секций
- Для работы с DB надо в lib и проект добавить [драйвер базы данных](#)
- Запустить `SqlStorageTest` (в `AbstractStorageTest` контакты и секции закоменченны), крденшелы к базе взять из `Config`
- Вынести общий код (`getConnection()`, `prepareStatement`, `catch SQLException`) в класс `SqlHelper`

 ichimax Update lesson14.md

25a6295 on 17 Mar

2 contributors  

31 lines (24 sloc) | 3 KB

Четырнадцатое занятие

Разбор Домашнего Задания-13

[см. коммит Lesson14 HW13](#)

1. JOIN

[см. коммит Lesson14 JOIN](#)

- [LEFT, RIGHT, INNER JOIN](#)
- [SQL Join \(youtube\)](#)
- Добавляем в `SqlStorage` контакты

2. Транзакции

[см. коммит Lesson14 Transaction_Batch](#)

- Транзакция. ACID. Уровни изоляции транзакций.
- Уровни изоляции транзакций в SQL
- Добавляем в `SqlStorage` транзакции
- Batch execute.

3. Установка/запуск Tomcat

- Скачать и установить [Tomcat 8](#). Устанавливать лучше простым копированием из архива в каталог (в том числе и для unix). Следите чтобы в пути не было пробелов и национальных букв.
- Для доступа к [Tomcat Manager](#) добавьте в конфигурацию Tomcat `TOMCAT_HOME\conf\tomcat-users.xml` права:

```
<user username="tomcat" password="tomcat" roles="tomcat,manager-gui,admin-gui"/>
```

- Запуск из `TOMCAT_HOME\bin\`: `catalina.bat start`

Домашнее задание HW14

- Закончить реализацию `SqlStorage` с контактами
- Выделить общие части ([DRY](#))

Пятнадцатое занятие

📺 Разбор Домашнего Задания-14

см. коммит Lesson15 HW14

📺 1. HTML, Tomcat

см. коммит Lesson15 web

- **Протокол HTTP**. Смотрим [демо приложение](#) в Chrome -> Инструменты разработчика
- Добавление в проект Web Facet. web.xml. Постороение/структура WAR. Статические ресурсы.
- Настройка и деплой в Tomcat. Tomcat manager.
- Запуск Tomcat из IDEA. Динамическое обновление без передеплоя.

📺 2. Сервлеты

см. коммит Lesson15 static_content

см. коммит Lesson15 servlets

- **Создаем Servlet**. Параметры. Кодировка.
- Дополнительно:
 - [How do servlets work?](#)
 - [Язык программирования Java: введение в сетевое программирование](#)
 - [Основы работы с HTML](#)
 - [Учебник HTML](#)
 - [Ссылки по HTML, JavaScript, CSS](#)

Домашнее задание HW15

- Сделать реализацию `SqlStorage.getAll` через 2 отдельных запроса: отдельно резюме и отдельно контакты.
- Добавить в реализацию `SqlStorage` и в базу секции (кроме `OrganizationSection`). Для `ListSection` склеиваем строки через `\n`.
- Сделать отображение таблицы резюме в сервлете.
 - [HTML таблицы](#)

 ichimax Update lesson16.md

e212e86 28 days ago

2 contributors  

38 lines (27 sloc) | 3.64 KB

Шестнадцатое занятие

Разбор Домашнего Задания-15. DB

см. коммит Lesson16 HW16 DB

Разбор Домашнего Задания-15. Servlet

см. коммит Lesson16 HW16 servlet

- Почему вы никогда не должны использовать MongoDB
- Жизненный цикл сервлета
- Tomcat maxThreads configuration
- Веб-приложение с Java Servlets

1. JSP

см. коммит Lesson16 jsp

2. include

см. коммит Lesson16 include

- Что такое JSP. Wiki JSP
- Predefined Variables in JSP
- Стандартные элементы action
- How to use relative paths in JSP
- Отличие Redirect от Forward

3. JSTL

см. коммит Lesson16 jstl

- JSTL
- JSTL для написания JSP страниц

4. CRUD

см. коммит Lesson16 crud

Домашнее задание HW16

- Доделать логику сервлета
- Дополнить отображение и редактирование JSP секциями

Семнадцатое занятие

Разбор Домашнего Задания-16

см. коммит Lesson17 HW16

Разбор Домашнего Задания-16 добавление резюме

см. коммит Lesson17 HW16_add_resume

1. Деплой в Heroku

см. коммит Lesson17 DB_credentials_as_resources

- Deployment with the Heroku CLI

2. Classloader

- Загрузка классов в Java
- Apache Tomcat Class Loader

3. Вебинар "Осваиваем Java String/JPA Enterprise"

- Слайды вебинара
- Из юниоров в разработчики: получаем первую работу
- Java Tools and Technologies Landscape Report 2016

Java Enterprise project Topjava

- Стажировка Spring/JPA Enterprise (Topjava)
- Ссылки по темам интервью, тестовое интервью
- Составление резюме, подготовка к интервью, поиск работы
- Как стать профессиональным Java разработчиком (Яков Файн)