کلاس ArrayList واسط List . java.util را پیاده سازی کرده

Students.remove(new Student("Ali")); ۰۸:۰۰

درست است می‌توان این متد هم شی ای ←

رو حذف کرد که می‌گرده دنبال Ali اگر بود ۰۹:۰۰

حذفش می‌کنه اگر نباشه هم خطا نمی‌ده

۱۰:۰۰

صرفاً هیچی حذف نمی‌کنه

اولین عضو این لیست برگردون

۱۱:۰۰

Student student = Students.get(0)

۱۲:۰۰

برخی از متدهای مهم ArrayList: ۱ int size()

۱۳:۰۰

← طول لیست برمی‌گرداند

۲ boolean isEmpty()   ۳ boolean Contains(object o)

۱۴:۰۰

← این لیست خالی است یا نه   ← مشخص می‌کنه لیست ما حاوی شی o است یا نه

۴ void add(E e) new E()   ۴ void remove(E e) new E() یا void remove(int index) ۱۵:۰۰

← یک عضو به لیست اضافه می‌کند   ← یک عضو از لیست حذف می‌کنه ۱۶:۰۰

۵ void clear()   ۶ <T> get(int index)

← همه اعضای لیست حذف می‌کنه   ← عضوی با این index برمی‌گرداند

۷ int indexOf(E e)   ۸ <E> set(int index, ~~set~~ E element)

← اولین موقعی که e در جدول دیده می‌شود. خانه بازگشت لیست است

← برای اینکه مشخص کنیم در خانه index دقیقاً کدوم شی قرار بگیره

۲۲ پنجشنبه
شهریور
۱۴۰۳

Collections. addAll

۰۸:۰۰

sort

reverseOrder()

۰۹:۰۰

binarySearch

۱۰:۰۰

copy

disjoint

۱۱:۰۰

Collectors. joining

۱۲:۰۰

محاسبه داده های آماری) summarizingDouble

groupingBy

۱۴۴۶ ربیع‌الاول ۸
2024 12 September

شهادت حضرت امام حسن عسکری (ع) (۲۶۰ هـ ق) و آغاز امامت حضرت ولی عصر (عج) (تعطیل)

PartitioningBy

toList

avaragingDouble

۲۳ جمعه
شهریور
۱۴۰۳

stream. map / mapToDouble

collect

۸. 7
هم میتونست filtering چند filter(t->t.startsWith("A")

sorted

avarage

limit

foreach iterate

۱۴۴۶ ربیع‌الاول ۹
2024 13 September

reduce

getAsDouble

distinct

of range

دیفالت equals اشاره گر ۲ شی را بررسی می کند

برای generate کردن hashcode و equals خود

IDE ها انجام می دهند فقط باید بگوییم روی کدام فیلدها

در VS code ← Right click →

Source Action

MAP یک Collection نیست ← فهرستی از اشیا را

یک MAP مانند یک جدول یا نگاشت از اشیا عمل می کند

که ستون اول key یکتا هستند (حافظه تکراری ندارد)

ولی Value می تواند تکراری باشد

یعنی مثلا MAP<String, Double> map

نگاشتی از رشته به عدد حقیقی است

MAP یک واسط و HashMAP یک کلاس از این واسط

که به جای add که در Collection ها بود اینجا put داریم

① put(,) ② get(key) ③ size() ④ isEmpty()

اگر Put(Key1 , Val1) Put(Key1, Val2) ← حافظه ها یکسان است چیزی که ذخیره می شود

Key1 است با Val2

در اصل put دوم update

چون آخرین Val است برای این key

java.util.map

Integer ⟵ int ⟸ auto_boxing ۰۸:۰۰

⑤ booleanContainsKey(Object key)

← مشخص می‌کنه که آیا این جدول در ستون اولش شامل ۰۹:۰۰
کلیدی مشخص شده هست یا نه ۱۰:۰۰

⑥ boolean ContainsValue(Object value) ۱۱:۰۰

⑧ void PutAll(map m) ⑦ <T> remove(key) ۱۲:۰۰

← همه اعضا m map را به MAP
اضافه می‌کنه (در اصل ادغام تو جدول) ⑨ Clear ۱۳:۰۰

⑩ Set<K> KeySet() ۱۴:۰۰

← مقادیر موجود در ستون اول به صورت یک
چون تکراری ندارن Set $\xrightarrow{\text{برمی‌گردونه Collection}}$ مجموعه Set برمی‌گرداند ۱۵:۰۰

⑪ Collection<V> values() ۱۶:۰۰

متد iterator() برای همه Collection ها وجود دارد

Iterator<Integer> ite = list.iterator()

ite.hasNext()   ite.next()

در واسط java.lang.Iterator تعریف شده

و واسط Collection از واسط Iterable ارث بری می کند

متد های واسط Iterator ⇐ ① iterator()

next() ③ hasNext() ②

remove() ④

مقدار آرایه

Arrays → .fill(array, 5) ①

آرایه

int[] Copy = Arrays.CopyOf(array, 10) ②

سایز آرایه

یک لیست غیرقابل تغییر

List<Integer> asList = Arrays.asList(1, 2, 3, 4) ③

تعدادی پارامتر برای رفتن توی لیست

asList(array)

تبدیل آرایه به یک لیست

④ sort(array) مرتب سازی یک آرایه با الگوریتم های سریع مرتب سازی

⑤ binarySearch(array, 7) اول sort بعد باینری سرچ

index برمی گرداند

indexOf و binarySearch مثل هم اند

Collections

① Collections.addAll(List, 1, 2, 3)

↲ اضافه کردن اینها به این لیست

② Collections.frequency(List, 1)

↲ تعداد تکرار این یارو در List

③ Sort(List)

↲ به ترتیب سورت میکنه List به کمک CompareTo ای که داخل اون کلاس اشیایی که داخل لیست هستن تعریف شده

Sort(List, Comperator)

↲ اگر ما Comperator هم به Sort پاس بدیم نیاز نیست اصلا List Comperable باشه یعنی متد ComperTo پیاده کرده باشه و از Comperable ارث بری کرده

④ reverse(List)

↲ عنصر اول آخر میاره و همینطور تا آخر

Subject.

Date.

Starter
① Create Spring boot Project

② ADD Dependencies :

① Spring web, ② Spring web services
③ Spring Boot DevTools ④ Spring Data JPA
⑤ H2 Database

③ Create Folders : ① Config ② entity
~~5~~ ⑤ service ③ Controller ④ repository

④ Config application.properties : add h2

⑤ ADD to main.java → @EnableJpaRepository (Folder)
@EntityScan (Folder)

آکادمی هواه

6) Creat classes For each Folder:

تغییر اسم فیلد json → @jsonProperty

1) entity → class → @Entity

@oneToone @table

@Column @Id

@joinColumn @generatedValue

@Transient @Temporal @Enumerated

Date Catalog

↳ validation

۲) repositories → interface extends JPARepo

<Book, Long

List<Book> findByName(String name)

۳) service → @service (@Component)

@Autowired

~~Privat~~ repository

hibernate است

(@PersistenceContext

Private EntityManager entityMana

@Transactional

↳ SpringFramwork

آکادمی هواه

Subject. Exception
Validation
key, value

Date. Ⓔ Controller → @Controller

value
headers
Produces
method
@RequestMapping("/api");

است است
{ { / , / ?
MediaType.
APPLICATION
XML

@Autowired
private service

@GetMapping
@ResponseBody

@RequestBody
(required=false) @PathVariable
HttpServletRequest
@RequestParam

getQueryString
getParameter
getParameterValues

Response Entity ولی ~~@ResponseTransfer~~
<HttpStatus
<Entitename>
<String>

@min @max
@valid @validated ؟

get Paramete map

JAXB → xml

Jackson → Json

Json → @JsonIgnore
@JsonProperty
@JsonFormat

Optional.orElseThrow(Custom class::new)

Exception
① Throw → Controller
② Custom class ← extends RuntimeException
③ ExceptionHandler
extends ResponseEntity
@ControllerAdvice
@ExceptionHandler

~~API~~ webservice

① یک اپلیکیشن Spring Boot چندتا API داره که API ها نوشته شده و برنامه‌ای قراره بنویسیم Call می کنه (باید بر اساس ادرس های API ها Call می‌شوند یکسری کارها انجام دهیم مثل فرانت در Database و

② Rest template. Delete کردن و ---)

که RestAPI روی اینترنت نه بیرون

فراخوانی شود و اطلاعات تشویق بدیم و