Теория за весь курс - повторение

Системы счисления

10 - ая: десятичная, десятиричная система счисления. 0, 1, 2, 3, ..., 9

Основание - 10, каждая цифра числа получается % 10 (деление с остатком). 54321 % 10 = 1 54321 / 10 = 5432

8 - ая: восьмиричная, 8, остатки от деления: 0, 1, 2, ... , 7 8 = 10

16 - ая: шеснадцатиричная, 16, 0, 1, 2, ... , 9 , A, B, С, D..., F

2 - ая: двоичная, 2 - основание, 0, 1 4 (10) = 100 (2)

Примитивные типы данных

int, byte, short, long, float, double, char, boolean

При объявлении каждой переменной, метод необходимо объявить ТИП.

Методы

* СИГНАТУРА метода
* название метода надо давать с маленькой буквы!!!
* если метод ничего не возвращает, то надо написать void (английское слово)
* конструктор - это ТОЖЕ МЕТОД

public int add(int a, int b) { // тело метода return a + b; }

Условие, тернарный оператор, switch case

if(),if() {} else {} помогает очень сильно сократить код и простое условие пример: ...

switch case - организует код, когда множественный выбор

Циклы - while, do while, for

Массивы

String[] word int[] numbers int[] arr

Как выглядит текущий элемент массива? - arr[i] Следующий за ним? - arr[i + 1] Предыдущий элемент? - arr[i - 1] Первый элемент? - arr[0] Последний? - arr[arr.length - 1]

arr.length - длина, кол-во элементов

Сортировка массива

* пузырьковая
* сортировка выбором
* ...

Они нужны, потому что БИНАРНЫЙ поиск в отсортированом массиве - это самый быстрый поиск O(log(n))

Принципы ООП

* инкапсуляция (все в классе ...)
* полиморфизм (перегрузка методов, разный набор аргументов ... )
* наследование (extends класса, implements от интерфейса )
* абстракция (абстрагируемся от конкретики, выделяем общие свойства объектов, берем только то, что нам надо...)

Что такое класс и т.д.

Абстрактные классы

* на базе абстрактного класса нельзя создавать конкретные объекты, объекты - только в дочерних классах

Интерфейсы

* только абстрактные методы (нет тела).

Классы-обертки

Integer, Byte, Character, Boolean, ... Расширяют методы для работы с такими данными.

Arrays, binarySearch

Arrays - это класс-обертка над array.

binarySearch - это метод, который возвращает индекс искомого элемента. Индекс > 0 - элемент найден Индекс < 0 - элемент НЕ найден, мог бы/должен стоять на -i - 1 месте.

Generics

, - параметры интерфейсов, тип данных может быть/должен быть определен на этапе создания объекта.

Comparable, Comparator

Comparable - для класса, задает метод compareTo, текущий сравниваем с присланным в метод. Желательно, чтобы это был естетсвенный (nativ) метод сортировки для объектов этого класса.

Компаратор - определяем на основе встроенного интерфейса в Java там, где нужен. Определяет метод compare (T o1, T o2).

Complexity

O(1) - константная, конечное кол-во операций. O(log(n)) - логарифмическая. O(n) - линейная, кол-во операций прямо пропорционально кол-ву элементов. O(n2) - квадратичная O(...)

JCF структуры

Iterable - hasNex(), next()

Collection - sort, CRUD-методы, ...

Списки - List, очереди - ..., Множества - Set

Списки: возможны дубликаты, есть индексы элементов, сортируемы, поиск и добавление - O(n) - линейная, не нужно следить за размером.

Множества: уникальность элементов, НЕТ индексов, HashSet - НЕ сортируется, поиск и добавление - O(1) - константная. TreeSet: уникальность элементов, НЕТ индексов, TreeSet - сортируется при наполнении элементами СРАЗУ, поиск и добавление - O(log(n)).

Мапы - это комбинация множества ключей и списков значений.

Java Time API

см. примеры: LocalDate.now - текущая дата ChronoUnits - дни, недели, полнедели, полмесяца, месяц, век и т.д.

Java Stream API

см. презентацию

Exceptions, try catch, throw

чаще практиковаться, не бояться.