Junior Java (Rus) Cheat Sheet

by SMF via cheatography.com/74584/cs/18740/

Java Data Types

boolean	true/ false
byte	8 bit
short	16 bit
int	32 bit
long	64 bit
char	16 bit (unicode)
float	32 bit (decimal)
double	64 bit (decimal)

Принципы ООП

Абстракция	Использование простых вещей для представления сложных с использованием объекта, классов и переменных.	Пример: описание должностей. Название должности - значимая инф-я, а описание - второстепенная.
Инкапсуляция	Это практика хранения полей в пределах класса private, а затем предоставление доступа к ним через открытые методы. Это защитный барьер, который сохраняет данные и код в безопасности внутри самого класса.	Пример: 10 методов для вычисления площади фигуры, но 9 из них скрыто. Используется только один ("найти площадь"), в котором вызываются остальные скрытые методы.

Принципы ООП (cont)

Наследование	Свойство системы, позволяющее описать новый класс на основе уже существующего с частично или полностью заимствующейся функциональностью.	Пример: наследование.
Полиморфизм	Свойство системы использовать объекты с одинаковым интерфейсом без инф-и о типе и внутренней структуре объекта.	Пример: общий интерфейс "Фигура" и реализации "Круг" и "-Треугольник". Метод "Нарисовать" один для всех фигур.

Object

public final native Class getClass()	возвращает в рантайме класс данного объекта.
public native int hashCode()	возвращает хеш-код
public boolean equals-(Object obj)	сравнивает объекты
protected native Object clone() throws CloneN-otSupportedException	клонирование объекта
public String toString()	возвращает строковое представление объекта.
protected void finalize() throws Throwable	вызывается сборщиком мусора, когда garbage collector определил, что ссылок на объект больше нет.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Object (cont)

public final native void notify()	просыпается один поток, который ждет на “мониторе” данного объекта.
public final native void notifyAll()	просыпаются все потоки, которые ждут на “мониторе” данного объекта.
public final native void wait(long timeout) throws InterruptedException	поток переходит в режим ожидания в течение указанного времени.
public final void wait() throws InterruptedException	приводит данный поток в ожидание, пока другой поток не вызовет notify() или notifyAll() методы для этого объекта.
public final void wait(long timeout, int nanos) throws InterruptedException	приводит данный поток в ожидание, пока другой поток не вызовет notify() или notifyAll() для этого метода, или пока не истечет указанный промежуток времени.

Все классы являются наследниками суперкласса Object. Это не нужно указывать явно. В результате объект Object о может ссылаться на объект любого другого класса.

Инициализация

Числа	0 или 0.0
Объекты (в том числе String)	null
char	\u0000
boolean	false

Определения

Класс	"Описатель" общих свойств группы объектов.
Поле, атрибут класса	Характеристика объекта.
Инициализация	Создание, активация, подготовка к работе, определение параметров. Выделение памяти под объект
Конструктор	Спец. метод, который вызывается при создании нового объекта. Инициализирует объект во время создания. Имя совпадает с именем класса.
Метод	Последовательность команд, которая вызывается по определенному имени. Функция, процедура.
Переопределение	Метод в классе-наследнике, совпадающий по сигнатуре с методом из родительского класса. Используется тогда, когда вы переписываете (переделываете, переопределяете) УЖЕ существующий метод.

Sponsored by **Readable.com**
Measure your website readability!
https://readable.com

Определения (cont)

Перегрузка	Создание метода с тем же именем, но с другим набором параметров. Использование одного имени, но с разными входными параметрами.
Интерфейс	Используется для создания полностью абстрактных классов. Создатель интерфейса определяет имена методов, списки аргументов и типы возвращаемых значений, но не тела методов.

Абстр. класс и интерфейс

Абстрактные классы используются только тогда, когда есть «is a» тип отношений	интерфейсы могут быть реализованы классами которые не связаны друг с другом.
Абстрактный класс может реализовывать методы	интерфейс может реализовывать статические методы начиная с 8й версии
С абстрактными классами вы теряете индивидуальность класса, наследующего его	с интерфейсами вы просто расширяете функциональность каждого класса.

Абстр. класс и интерфейс (cont)

класс может наследоваться (реализовывать) только от одного абстрактного класса	класс может наследоваться (реализовывать) от многих интерфейсов

Интерфейс может описывать константы и методы. Все методы интерфейса по умолчанию являются публичными (public) и абстрактными (abstract), а поля — public static final. С java 8 в интерфейсах можно реализовывать default и статические методы.

Ключевые слова

Сколько ключевых слов зарезервировано языком?

> 50, два из них не используются: const, goto;

Примитивы

> (byte, short, int, long, char, float, double, boolean)

Циклы и ветвления

> (if, else, switch, case, default, while, do, break, continue, for)

Исключения

> (try, catch, finally, throw, throws)

Области видимости

> (private, protected, public)

Объявление \ Импорт

> (import, package, class, interface, extends, implements, static, final, void, abstract, native)

Создание \ Возврат \ Вызов

> (new, return, this, super)

Многопоточность

> (synchronized, volatile)

Другие

> (instanceof, enum, assert, transient, strictfp, const, goto)

Sponsored by Readable.com
Measure your website readability!
https://readable.com