<https://www.youtube.com/watch?v=P-_Nzi_mCRo>

Java Main Method Explained - What Does All That Stuff Mean?

-JRE has to call your main method

Public

-this method has to be callable by something outside of this class

Static

-JRE doesn’t want to create an object to call main method

<https://www.youtube.com/watch?v=1XAfapkBQjk&t=1s>

Exception Handling in Java Tutorial

Exception

-Unwanted event that interrupts the normal flow of your program

Throwable

-Errors and Exception

-Not recommended to catch Errors

Multicatch Statements



A screen shot of a computer code

Description automatically generated

Finally

-will always run exception or not even if return statement inside try

-will return 5, return in finally will override return in try

<https://www.youtube.com/watch?v=bCPClyGsVhc>

Checked vs. Unchecked Exceptions in Java Tutorial - What's The Difference?

Checked

-Java makes you deal with it during compile time

-Can handle it with a try/catch or throws

A screen shot of a computer program

Description automatically generated

Throws

-Tells Java that I know this code can throw an exception but I don’t want to deal with it

-the code that calls it will deal with it

-So error on readFile, because now that code needs to deal with it. Could even throw it there, won’t throw an checked exception where on main

-Should deal with uncheck exception

A yellow text on a black background

Description automatically generated

<https://www.youtube.com/watch?v=AoUVdLWLFQw>

.equals() vs. == in Java - The Real Difference

-Use == for primitives but Strings will work because BTS of String literals (not using new command)

-java will use interning and have the same reference point to the same String to save memory

A green and orange text

Description automatically generated A screen shot of a computer

Description automatically generated

-If you don’t implement equals, it will use Object equal method.

<https://www.youtube.com/watch?v=NbYgm0r7u6o>

Array vs. ArrayList in Java Tutorial - What's The Difference?

<https://www.youtube.com/watch?v=jhDUxynEQRI>

Java Polymorphism Fully Explained In 7 Minutes

-Overriding and Overloading

<https://www.youtube.com/watch?v=r9CMJZ4T__8>

Final Keyword in Java Full Tutorial - Final Classes, Methods, and Variables

A computer screen with text

Description automatically generated

-This is okay, just can assign it one time

A screen shot of a computer

Description automatically generated

-To make a global variable

<https://www.youtube.com/watch?v=oqPiEc2zNb0>

What is a POJO in Java? Almost EVERYONE Gets This Wrong

POJO

-can’t extend (can’t be a child class)

-can’t implement any interfaces

-no outside annotations

Basically don’t need any outside libraries to use it

Java Bean

-Have No-args constructor

-Properties must be private

-Getters/Setters

-Serializable (implements Serializable interface). Tells Java this can be written to database and files

Misc

@Entity need JPA/Hibernate to use

DTO

In my experience a DTO is basically a simple class, usually a POJO, whose job is to work as a middle-man between two representations. For example, you might have a web service request with a certain class structure, and also a database entity with a similar, but separate class structure. To avoid tightly coupling those two classes together, we typically will have a DTO (data transfer object) class that essentially holds the same data, and create mappings from the request to the DTO, and from the DTO to the database entity. And perhaps mappings for the other way around too, if you also want to be able to return those database entries to the user.

So a DTO is more of a name for the purpose of the class, rather than aspects of what the class has or doesn't have like a pojo or javabean.

<https://www.youtube.com/watch?v=5dscMs2hnDI>

LinkedList vs ArrayList in Java Tutorial - Which Should You Use?

They are similar because both of them are part of List Interface

ArrayList: uses an Array

LinkedList: uses nodes

<https://www.youtube.com/watch?v=H62Jfv1DJlU&t=5s>

Map and HashMap in Java - Full Tutorial

-putIfAbsent

<https://www.youtube.com/watch?v=BwkmIXjWWJc>

Inner Class Java Tutorial - Creating and Using Inner Classes







Make the class static if you don’t want to create an outer class instance

A screen shot of a computer code

Description automatically generated

Method local inner class

SKIP java Anonymous Inner Class

<https://www.youtube.com/watch?v=r_MbozD32eo>

Multithreading in Java Explained in 10 Minutes

-Multiple Different path of code at the same time

Two ways of creating a new thread

-extends Thread

@Override run()

A black background with yellow and white text

Description automatically generated

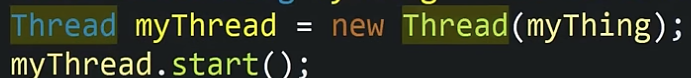
-To create 5 threads

-If one thread fails, it won’t impact the others

OR

-implement Runnable

@Override run() (samething?)



-Can’t call start anymore because we don’t extend Threads anymore

So Which one?

Thread

-you don’t have to create Thread object

Runnable

-Implement interface so you can extend another class

A black background with blue and green text

Description automatically generated

-If you want to wait for the thread to complete

<https://www.youtube.com/watch?v=-5NC5_sI-vQ>

Java is ALWAYS Pass By Value. Here's Why

<https://www.youtube.com/watch?v=HvPlEJ3LHgE>

Abstract Classes and Methods in Java Explained in 7 Minutes

-Can’t create objects

-enforces what sub classes has to have

Abstract Methods

-no implementation, your sub class will implement or make them abstract

Vs Interface

-implements vs extends

-Interface fields are static and final, must instantiate

-Abstract can have methods defined

<https://www.youtube.com/watch?v=pgBk8HC7jbU>

Java Constructors - Full Tutorial

<https://www.youtube.com/watch?v=HpuH7n9VOYk>

Upcasting and Downcasting in Java - Full Tutorial

Upcasting

-take an object and cast it to its superclass type

Downcasting

-take an object and cast it to its sub types

Upcasting



InstanceOf

<https://www.youtube.com/watch?v=wq9SJb8VeyM>

Java Enums Explained in 6 Minutes

-Predefined set of values that don’t change





<https://www.youtube.com/watch?v=ETLHbHenW44>

This Keyword in Java Full Tutorial - How to Use "this"

<https://www.youtube.com/watch?v=Qb_NUn0TSAU>

Super Keyword in Java Full Tutorial - How to Use "super"

-use super to super methods instead of child class

-can only be used in the child class

-can not access private of super

-super constructor to reduce code duplication

<https://www.youtube.com/watch?v=K1iu1kXkVoA>

Generics In Java - Full Simple Tutorial

<T> is called type parameter



-Can store everything but not type safe



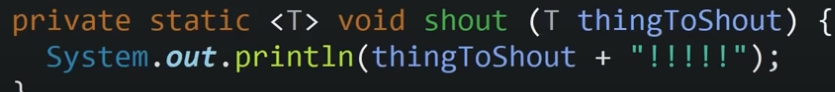
Only classes that extends Animal can use it



-Still use extends for interfaces



-Use & but can only have one class and list it first



Generic Method





-Wildcard



-Bound it

<https://www.youtube.com/watch?v=tj5sLSFjVj4>

Lambda Expressions in Java - Full Simple Tutorial

SKIP

<https://www.youtube.com/watch?v=gJ9DYC-jswo>

Records In Java - Full Tutorial - The Best New Java Feature You're Not Using

-Avoid boiler plate in certain classes



-Without record, classes that carry data will want to have a contructor, hashcode, equals, toString

Record

-special type of class like Enum

-good for holding and carrying Data

-final classes (can’t be extended by other classes)

-generates private/final fields for the components, getters (but no get, just name()), canonical constructor (overridable, good for validation Ex employee number cant be negative), toString, hashcode, equals (allowed to override)

-can create instance/static methods and static fields (also final), CANNOT create instance fields has to be in the list of components

-can’t extend but allowed to implement interface

-Compact Constructor (unique to Record)

A black background with white text

Description automatically generated

<https://www.youtube.com/watch?v=DkZr7_c9ry8>

Annotations In Java Tutorial - How To Create And Use Your Own Custom Annotations

Skip

Asian girl Pink dress: 44:35 ass slap harder

515- 1 O in first two scenes, CowG: your so big, O 22:10, 24:46 nah, ? just before POV: so deep