Plain Old Java Object

A plain old Java object (whose acronym is **POJO**) is a class that generally only has instance fields.

It's used to house data, and pass data, between functional classes.

It usually has few, if any methods other than getters and setters.

Many database frameworks use POJO's to read data from, or to write data to, databases, files or streams.

Examples of POJOS

A POJO also might be called a bean, or a JavaBean.

A JavaBean is just a POJO, with some extra rules applied to it.

A POJO is sometimes called an Entity, because it mirrors database entities.

Another acronym is DTO, for Data Transfer Object.

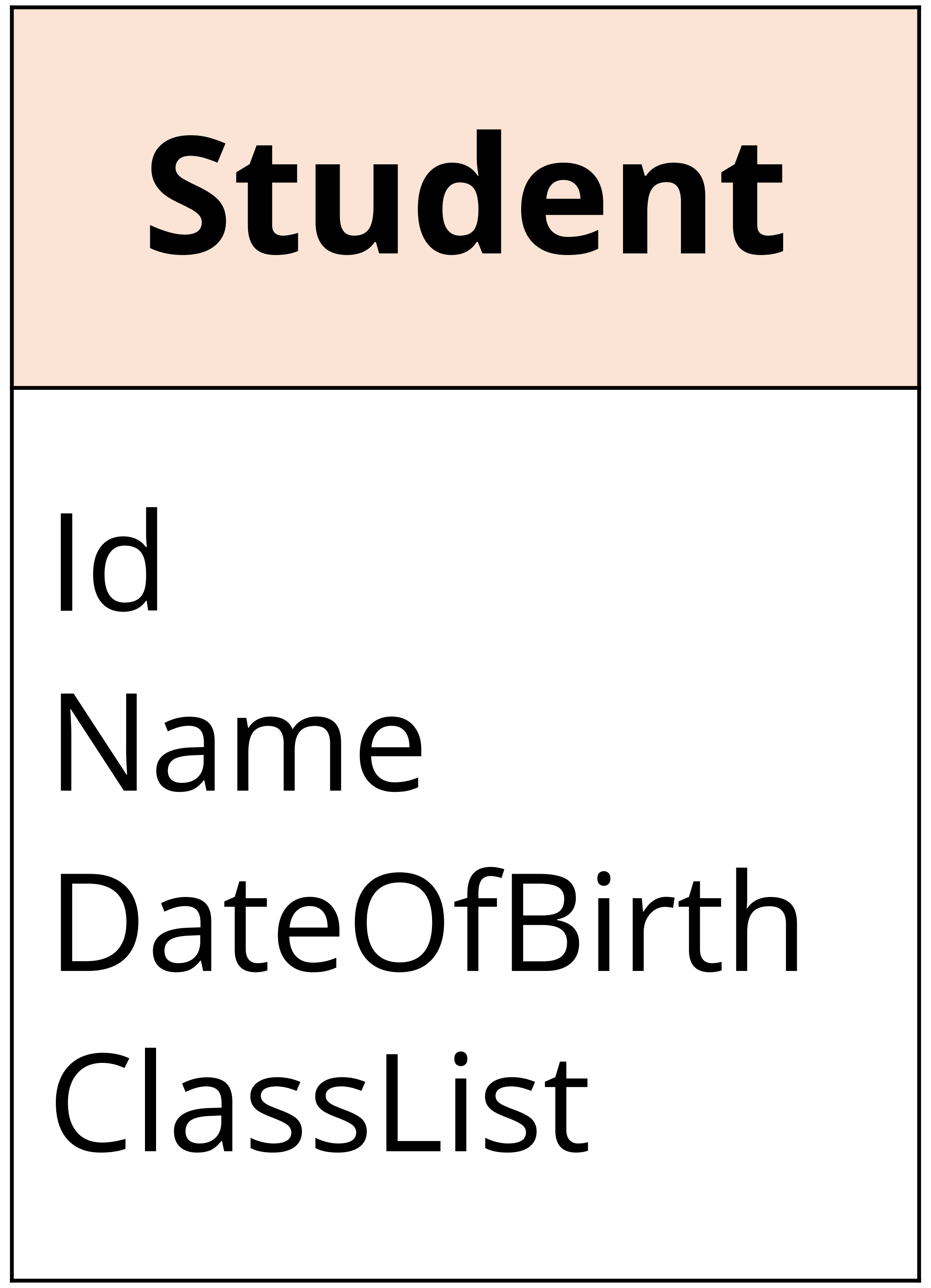
It's a description of an object, that can be modeled as just data.

Support for POJO creation

There are many generation tools, that will turn a data model into generated POJO's or JavaBeans.

You've seen an example of similar code generation in IntelliJ, which allowed us to generate getters, setters, and constructors in a uniform way.

The Entity - The Student Table



Student

Id
Name
DateOfBirth
ClassList

Annotation

Annotations are a type of metadata.

Metadata is a way of formally describing additional information about our code.

Annotations are more structured, and have more meaning, than comments.

This is because they can be used by the compiler, or other types of pre-processing functions, to get information about the code.

Metadata doesn't effect how the code runs, so this code will still run, with or without the annotation.

Overridden Method

An overridden method, is not the same thing as an overloaded method.

An overridden method is a special method in Java, that other classes can implement, if they use a specified method signature.