

## 8.2.

Genannte Beispiele aus der Technologievorlesung:

Singleton: MovieManagerModel

Factory Method: SearchMasterFragment(Creator), MovieManagerActivity(Client)

Builder: CalenderConstraints

Wrapper: RecyclerView.Adapter

Observer: setPositiveButtonListener

Definitionen:

Singleton: Ermöglicht dass nur ein Objekt erzeugt wird und global verfügbar ist (z.B. private static classname INSTANCE)

Factory Method: Ermöglichen erzeugen von Objekten an Unterklassen zu delegieren (z.B. onCreate(), newInstance())

Builder: Trennung der Konstruktion eines Objekts von seiner Darstellung (z.B. .Builder())

Wrapper: Schnittstelle wird angepasst, damit Klassen einfacher zusammenarbeiten können (z.B. Adapter)

Observer: Ermöglichen Erhalt von Informationen über Änderungen anderer Objekte (z.B. ButtonListeners)

Singleton:

Storage.StorageManagerAccess

Singleton: private static final StorageManagerAccess INSTANCE = new StorageManagerAccess();

Factory Method:

Ui.master.comparator.DataMasterFragment

Factory Method: public static DataMasterFragment<Movie> newMovieFragmentInstance(@StringRes int nameld, Set<Movie> movies)

Builder:

Ui.master.comparator.CategorizedComparator

Builder: public interface CategorizedComparator<T> extends Comparator<T>

String getCategoryNameFor(T object);

Wrapper:

Ui.adapter.SortingCriteriaAdapter

Wrapper: public SortingCriteriaAdapter(Context context, List<SortingMenuItem<T>> items)

Observer:

Ui.dialog.PerformerSafeRemovalDialog

Observer: .setPositiveButton(R.string.yes, (dialog, which) ->  
positiveListener.accept(performersToRemove))