

# Aufgabe 8.1 – Entwurfsmuster

## Singleton

class RuntimeStorageAccess

static getInstance(): `public static RuntimeStorageAccess getInstance() {return INSTANCE;}`

static INSTANCE: `private static final RuntimeStorageAccess INSTANCE = new RuntimeStorageAccess();`

## Factory Method

Class PortrayableMasterFragment:

`public static PortrayableMasterFragment<Movie> newMovieFragmentInstance`  
`public static PortrayableMasterFragment<Performer> newPerformerFragmentInstance`

## Builder

PortrayableRVAdapter.<Movie>buidler: Builder

MovieMasterFragment: Director

PortrayableRVAdapter<Movie>: Product

From MovieMasterFragment

```
protected PortrayableRVAdapter<Movie> createAdapter() {  
    String constraint = filter.getText().toString();  
  
    adapter = PortrayableRVAdapter.<Movie>buidler()  
        .setTaskExecutor((OneTimeTaskExecutorRudiment) getActivity())  
        .setHost(this)  
        .setOrders(orders)  
        .setModelData(originalData)  
        .setConstraint(constraint)  
        .setDetailActivityGetter(() -> MovieDetailActivity.class)  
        .setDetailEditActivityGetter(() -> MovieDetailEditActivity.class)  
        .setRemoveFromStorage(STORAGE::removeMovie)  
        .setAfterUpdate(result -> afterUpdate())  
        .setAfterRequestedCreation(this::afterRequestedCreation)  
        .setCanDelete(this::canDelete)  
        .build();  
  
    return adapter;  
}
```

## Observer

Interface HeaderListener

onClickListener

## Wrapper

SimpleAdapter

```
public void onBindViewHolder(@NonNull SimpleViewHolder<T> holder, int position) {holder.bindView(position, binder);}
```

SimpleViewHolder erbt:

```
public class SimpleViewHolder<T> extends ViewHolder
```

## Template Method

### Class PortrayableMasterFragment

Abstrakte Methode createOrders() wird in der Methode onViewCreated() aufgerufen

```
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {  
    super.onViewCreated(view, savedInstanceState);  
    AutoBind.bindAll(this, view);  
  
    createOrders();  
    setupRecyclerView();  
  
    orderButton.setOnClickListener(this::openOrderPopupMenu);  
    filter.addTextChangedListener(createOnTextChangedListener(getAdapter().getFilter():filter));  
    addButton.setOnClickListener(v -> getAdapter().createObject());  
}
```

...

```
protected abstract void createOrders();
```