**Design Patterns**

Group members:

Alex Mednov

Gabriel Guevara Lopez

Contents

[Factory 3](#_Toc163412794)

[Benefits: 3](#_Toc163412795)

[Negatives: 3](#_Toc163412796)

[Builder 3](#_Toc163412797)

[Benefits: 3](#_Toc163412798)

[Command 4](#_Toc163412799)

[Benefits: 4](#_Toc163412800)

[Drawbacks: 4](#_Toc163412801)

# Factory

The factory method could be implemented in the creation of slides. One of the benefits of factory method is that it is possible to use to create different elements. The content of the slide is undetermined; thus, a factory method could be used to create them in a better way. It would also benefit the resource consumption, as it would decrease them.

## Benefits:

* Lessens coupling.
* The creation code is going to be concentrated on a specific point in code.

## Negatives:

* Complicates the code.

# Builder

Builder will be implemented in the creation of slides. Builder will act as an interface for AnimationBuilder, BaseLineBuilder and VideoBuilder and as last a Director who will be in charge of creating the slides.

The Builder pattern simplifies object creation by separating the construction of complex objects from their representation, making it easier to handle objects with many optional components and providing flexibility in construction through various Builder implementations.

## Benefits:

* Building objects one by one and
* Lets you use the same code to build different versions of products.
* Isolating complex construction code from the business logic.

# Command

The Command Pattern is a behavioral design pattern that encapsulates requests into objects. In our project, the Command Pattern facilitates the implementation of various slide manipulation commands, such as Next, Previous, Exit, New, Save, GoTo, Help, Invoker, Reciever, Open, and a Command interface enhancing flexibility in handling user interactions with slides.

## Benefits:

* Loose coupling by separating the sender of a request from its receiver, allowing for flexibility in adding new commands without altering existing code.
* You can implement undo/redo.

## Drawbacks:

* The code may become more complicated since you’re introducing a whole new layer between senders and receivers.

**Other stuff**

Other stuff includes naming conventions, coding conventions, some code is repeated multiple times, some variables are repeated multiple times and are not used, getters and setters are not there. As well as some code needs to be rearranged and to be made cleaner.