**Design patterns**

**Factory**

The factory method could be implemented in the creation of slides. One of the benefits of factory method is tha it is possible to use to create diffrent 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 recource consumption, as it would decrease them.

**Benefits:**

Lessens coupling

The creation code is going to be concentrated in a specific point in code

**Negatives:**

Complicates the code

Perhaps switch it for the builder, but we need to ask for that.

**Composite**

As there are going to be multiple slides in a single presentation, it would be beneficail to use composite pattern to ease the work with them. By using composite, instead of dealing with ludicrous amount of slides, it would be possible to deal with a single class, that is going to represent them all.

**Benefits:**

Allows to easily work with big structures of slides

**Negatives:**

Difficult to implement for absolutly diffrent classes (does not apply to us, as we will be using to describe slides)

**Decorator**

Would allow for easy addition of new functionality without breaking existing one, could be beneficial for slides, when new type of slides are to be created. (where you can watch a video for example)

**Benefits:**

Expand object behavior without making new subclass

**Negatives:**

Not easy to remove a wraper from a wraper stack (should not apply to us)

Code can be a bit ugly.

**Other stuff**

Other stuff includes naming conventions, coding conventions, some code is repeated multiple times, some variable are repited multiple times and are not used, geters and seters are not there.