# Changes

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Architectural Problem** | **Proposed Solution** | **Reasoning** |
| 1 | BitMapItem, SlideItem, TextItem have unused constructors | Remove unused code | Improve code readability, declutter code |
| 2 | setSlideNumber() method in Presentation class for out of bounds slide number values. If the value input is higher than the number of slides, it will update the counter, but keep the same slide on. If the value input is <1 the slide becomes blank. | Introduce an out of bounds slide number check before updating currentSlideNumber | Fix a bug |
| 3 | getCurrentSLide() in Presentation accesses the currentSlideNumber field directcly instead of using a getter. | Use a getter method getSlideNumber() | Proper usage of access levels improves security of the class. |
| 4 | Accessor is an abstract class with unused fields and only two methods. Its inherited classes do not obey the Liskov Substitution Principle. It makes more sense for Accessor to be an interface. | Make Accessor an interface and remove unused fields. Update its implementation inside XMLAccessor and DemoPresentation. | The relationship will be more understandable now. It will be easier to add any new accessor classes, improving adaptability of the application. |
| 5 | Style class has a field “styles”, which is an array that contains instances of the Style class itself. | Move the styles array, and the static getStyle() method into JabberPoint class as styles of slides are application wide and it makes sense to initialize them when the application starts up. Also, all presentations use the same styles. | Remove circular dependency. Make architecture less confusing. Improves testability as styles can be changed easier and tested with users. |