3. We could have best implement the Stack class by using generic types, instead of hard defined type (like Integer). I would allow to avoid any changes in a lot a cases, making the object much more flexible.

The only difference would be in the types, not in the global architecture.

4. All tests would not be changed, only the ones using a strongly defined type, like head() or push(), or constructor taking an array of int.

5.

9. After this implementation, a lot of tests can be avoided. In fact, the only tests really needed are the ones for « push », « pop », « head », « back » in order to check the algorithm, and the constructors.

Some methods are exactly the same as the ones in the Stack class but used in the Queue context, which is nearly identical (and in fact is identical for those methods) making new tests useless.

10.

11. As told in question 9, we reused some methods we had already implemented in the Stack class. To avoid that, we could have used an abstract class « StorageClass », having those methods, allowing us to best implement the reuse.