

## Практическая работа № 4

*Вы сдали уже три работы. Вы смирились. Вы познали дзен. Теперь вам ничего не стоит переписать её с нуля опять! Это всё равно будет легче, чем настраивать вторую работу.*

*Постойте... если придётся делать заново, то и БД подключать надо заново...*

### ModelViewPresenter

Необходимо спроектировать приложение согласно правилам (рекомендациям) архитектурного решения MVP. Функционал системы остается прежним. Итак, нужно доработать, а лучше переписать с чистого листа, лабораторную работу №3 следующими образом:

**Model** должна быть представлена:

- доменными классами;
- классами, реализующими бизнес-логику приложения;
- интерфейсами, которые позволят реализовать принцип инверсии зависимостей в классе **Presenter**, и дать ему работать с классами бизнес-логики через переменные интерфейсного типа;

**Presenter** представлен классами, которые "обслуживают" представление (**View**). То есть подписаны на его события и занимаются перерисовкой **View** путем вызова соответствующих методов (методы должны быть объявлены в интерфейсе, который имплементирует **View**). **Presenter** "общается" с **Model** и слушает ее события о проделанной работе классом бизнес-логики.

**View** генерирует события, на которые подписан **Presenter**, а также занимается отрисовкой данных.

**Shared**. Вспомогательная библиотека, которая позволит избавиться от циклических ссылок в нашем решении. Библиотека содержит интерфейс представления (**IView**), на который ссылается сборка **Presenter** и **View**. Таким образом, архитектуру Вашего приложения можно условно представить следующим образом (рис. 1):

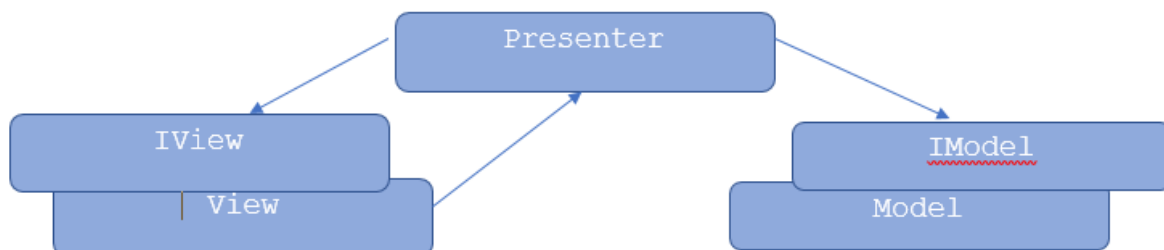


Рисунок 1 – Типа архитектура для этой работы