**Лабораторная №**4  
Just Patterns

Вступление.

В разработке любого ПО одним из важнейших элементов являются “Patterns” – шаблоны проектирования. Эти шаблоны представляют собой наиболее часто используемые архитектурные решения взаимосвязей объектов между собой.

Шаблоны проектирования – большая тема ООП. Правильное понимание многих шаблонов приходит только с долгой практикой. Без базовых навыков ООП применить их на практике очень сложно.

Но тему надо знать. Один из частых вопросов на собеседовании, особенно джуна – «Какие паттерны ты знаешь и какие использовал?»

Давайте познакомимся с шаблонами, чтобы каждый знал «***в каком ящике искать нужный инструмент***», когда понадобится.

**Пункт №1 - ознакомиться с паттернами MVP, MVC, MVVM**

***<https://habr.com/ru/post/215605/>***

Эти шаблоны созданы для разделения программного проекта на модули в целом. Т.е. они накладывают базовую архитектуру приложения. Они придуманы, чтобы отделить работу интерфейса от основной бизнес-логики приложения. Когда вы выбираете в VIsual Studio проект “Windows Forms” (mvp), “WPF” (mvvm) или проект Android (mvc) приложения, вам предлагается сразу работать по одному из этих паттернов. В раздельных частях проекта описывается интерфейс и логика. И первое преимущество этих шаблонов для командной разработки в том, что можно параллельно и независимо разрабатывать части программы.  
Интерфейс будет работать в одном потоке, а логика в другом( или других). Это дает программе, если на уровне логики (model) произошла ошибка или она зависла, отобразить для пользователя соответствующее сообщение или отрисовывать скроллбар(загрузки например). Это успокаивает нервозных юзеров, любящих долбить по кнопкам, нервно вызывая диспетчер задач, и не давая программе времени подумать и отвиснуть.

Ознакомление с этой темой упростит дальнейшее углубленное изучение разработки ПО, и даст базовое представление об архитектуре и взаимодействии слоёв операционных систем и ПО.

**Пункт №2**

Индивидуальный паттерн по вариантам. Номер варианта соответственно фамильному списку. После 10-го счет сначала.

Нижеперечисленные паттерны не описывают общую архитектуру проекта, а задают хитрую, гибкую логику фрагменту программного проекта. Упрощают использование программного кода, исключают его дублирование, или\и задают особое поведение выполнения программы.

Подготовить доклад о шаблонах проектирования, включающий общие определения, понятия темы, и индивидуальный пример, TimesNewRoman 14 - 4-5 страниц. Про UML и примеры в с++ можно почитать в [этой доп. лекции](https://docs.google.com/presentation/d/1-jRIZSzACFHWDPl-CAvMzZoPH8rQv1Gz/edit?usp=sharing&ouid=111716004947648948679&rtpof=true&sd=true)

*1) Адаптер*

*2) Прототип*

*3) Фасад*

*4) Фабрика*

*5) Строитель*

*6) Компоновщик*

*7) Наблюдатель*

*8) Стратегия*

*9) Синглтон*

*10) Итератор*

**1. Найти инф-ю о паттерне в интернете и назвать его тип (***порождающий, структурный, поведенческий***).  
2. Найти UML-диаграмму, описывающую данный паттерн**

**3. Привести “жизненный пример”, как работает данный паттерн.**

**Большинство примеров лучше всего искать в работе компьютерных игр.**

<https://tproger.ru/translations/design-patterns-simple-words-1/> - неплохая статья на русском

<https://refactoringguru.cn/ru/design-patterns/cpp>

- с примерами на плюсах