Министерство образования Республики Беларусь

Учреждение образования «Столинский государственный аграрно-экономический колледж»

Учебная практика «Разработка и сопровождение программного обеспечения»

Занятие № 19

Номер учебной группы П-16

Фамилия, инициалы учащегося Флорьянович В. Л.

Дата выполнения работы 30.11.22

Тема работы: «Использование шаблонов при разработке программного

Лучшим шаблоном для моего проекта является MCV, т.к. данный шаблон используется в формах: Кнопки, ввод текста – контроллеры; Методы и функции – методы; Таблицы, вывод текста – Вид.

В моём проекте в контроллеры будут входить: Кнопки «Регистрация», «Вход», «Создать новую тему», «Создать автоматически», «Выберите сложность», «Добавить вопрос», «Вперёд», «Назад», «Изменить», «Сохранить изменения», «Множественный/Одиночный выбор», «Выход», «Завершить тест», «Выбрать ответ»; Места ввода текста; Круглые и квадратные кнопки ввода (как в тестах).

В Методы будут входить: Функции «поиска индекса пользователя», «соответствия пароля», «добавления нового пользователя (Регистрации)», «создания теста», «добавления вопроса», «сохранения изменений», «навигации», «изменения кол-ва выборов ответа», «сохранения ответов», «оценки», «выхода».

В Вид будут входить: выводимый текст (Вопрос, ответы, наименования, уточнения), диалоговые окна ошибок.

Ответы на вопросы:

Шаблон проектирования или паттерн (англ. design pattern) в разработке программного обеспечения — повторяемая архитектурная конструкция, представляющая собой решение проблемы проектирования в рамках некоторого часто возникающего контекста.

В сравнении с полностью самостоятельным проектированием, шаблоны обладают рядом преимуществ. Основная польза от использования шаблонов состоит в снижении сложности разработки за счёт готовых абстракций для решения целого класса проблем. Во-первых, слепое следование некоторому выбранному шаблону может привести к усложнению программы. Во-вторых, у разработчика может возникнуть желание попробовать некоторый шаблон в деле без особых оснований

* Делегирование
* Функциональный дизайн
* Интерфейс
* Абстрактная фабрика
* Строитель
* Фабричный метод
* Прототип
* Наблюдатель

Объект внешне выражает некоторое поведение, но в реальности передаёт ответственность за выполнение этого поведения связанному объекту.

Гарантирует, что каждый модуль компьютерной программы имеет только одну обязанность и исполняет её с минимумом побочных эффектов на другие части программы.

Общий метод для структурирования компьютерных программ для того, чтобы их было проще понять.

Класс, который представляет собой интерфейс для создания компонентов системы.

Класс, который представляет собой интерфейс для создания сложного объекта.

Определяет интерфейс для создания объекта, но оставляет подклассам решение о том, какой класс инстанцировать.

Определяет интерфейс создания объекта через клонирование другого объекта вместо создания через конструктора.

Определяет зависимость типа «один ко многим» между объектами таким образом, что при изменении состояния одного объекта все зависящие от него оповещаются об этом событии.