

Паттерны программирования
Лабораторная работа № 4. MVC

Каждое задание должно быть загружено на личный git-репозиторий отдельным коммитом. Лабораторная работа выполняется в одной папке. Защита работы возможна на любой лабораторной работе от 1 до 16. Каждое из 4 заданий проверяется отдельно с учетом вопросов преподавателя. Задание засчитывается отдельно, лабораторная работа зачтена в случае выполнения всех 4 заданий.

Если часть задач выполнена в один коммит, работа не проверяется. Если все коммиты сделаны в один час, работа не проверяется.

Задание 1 Реализация Create.

Задачи

1. При нажатии на кнопку создать открывается Модальное окно, содержащее все поля класса Student, кроме ID, у этого окна свой контроллер. При возврате в контроллер основного окна происходит отображение элементов.
2. В окне Student происходит валидация заполнения обязательных полей и корректного заполнения полей строчками, ТОЛЬКО при прохождении валидации кнопка «Ok» доступна. При нажатии на кнопку отмена закрытие окна и его контроллера, при нажатии на кнопку Ok вызов метода контроллера, создание объекта и добавление объекта по соответствующему методу класса Student_List.
3. Построить диаграмму последовательность прецедента Создать сущность студент.
4. Протестировать работу на БД и файле.
5. Внести изменения в диаграммы классов.

Вопросы.

- a. Опишите принципы работы с модальным окном.
- b. Покажите и объясните изменения в диаграммах

Задание 2. MVC – CRUD - update. * (НЕОБЯЗАТЕЛЬНО)

Задачи.

1. При нажатии на кнопку Обновить открывается Модальное окно, описанное в предыдущем пункте, содержащее все поля класса Student, кроме ID, у этого окна свой отдельный контроллер,

- возможно изменить только ФИО. При возврате в контроллер основного окна происходит отображение элементов.
2. Внести коррективы в класс модального окна так, чтобы разные контроллеры вызывали одинаковые методы у класса окна, при этом необходимые поля были заполнены, необходимые поля доступны или недоступны для заполнения.
 3. При нажатии на кнопку отмена закрытие окна и его контроллера, при нажатии на кнопку Ok вызов метода контроллера, редактирование существующего объекта и изменение объекта по соответствующему методу класса Student_List.
 4. Построить диаграмму последовательность прецедента Изменить сущность студент.
 5. Протестировать работу на БД и * (НЕОБЯЗАТЕЛЬНО) файле.
 6. Внести изменения в диаграммы классов.
 7. * (НЕОБЯЗАТЕЛЬНО) Добавить на главном окне обработку кнопок Добавить гит, добавить контакты и отдельное отображение студентов, не прошедших валидацию по наличию контактов или гита. Добавить обработку новых кнопок согласно паттерна MVC и в рамках одной иерархии классов(один класс окна, 4 класса контроллера – новый объект, изменения ФИО, гита, контактов).
 8. * (НЕОБЯЗАТЕЛЬНО) Изменить диаграмму последовательность прецедента Изменить сущность студент.
 9. * (НЕОБЯЗАТЕЛЬНО) Протестировать работу на БД и *(НЕОБЯЗАТЕЛЬНО) файле.
 10. * (НЕОБЯЗАТЕЛЬНО) Внести изменения в диаграммы классов.

Задание 3. Фабрика. * (НЕОБЯЗАТЕЛЬНО)

Задачи

1. * (НЕОБЯЗАТЕЛЬНО) Разобрать паттерн фабрика. Написать тривиальный пример такого паттерна.
2. * (НЕОБЯЗАТЕЛЬНО) Внедрить паттерн фабрика для построения объектов нового окна и состояния так, чтобы ВСЯ логика записи ассоциированных классов в поля друг друга происходила внутри этой фабрики, но НЕ в конструкторах и НЕ в контроллере основного окна.
3. * (НЕОБЯЗАТЕЛЬНО) Применить паттерн фабрика для основного окна и его контроллеров, для этого придётся выделить супер классы сущностей и списков сущностей с одинаковым интерфейсом. Реализацию туда пока выносить не нужно, будет сделано в последующих лабах.

Вопросы.

- a. Опишите, в каком случае рекомендуется применять паттерн Фабрика.
- с. Напишите на бумаге пример такого паттерна.

Задание 4. Удаление.

Задачи

1. Реализовать обработку кнопки удалить. По нажатию на кнопку вызывается метод контроллера, метод смотрит объекты, считающиеся выделенными в классе `Data_list_student_short`, получает список ID таких элементов и удаляет элементы по ID, вызывая соответствующий метод класса `Student_list`
2. Построить диаграмму последовательность прецедента Удалить сущность студент.
3. Протестировать работу на БД и файле * (НЕОБЯЗАТЕЛЬНО).
4. Внести изменения в диаграммы классов * (НЕОБЯЗАТЕЛЬНО).