

Техническое Задание: Консольное Приложение "Каталог Отелей"

1. Общие сведения

| Параметр | Значение |
|------------------------|----------------------------------------------------------------------------------------------------------------|
| Название | Каталог и Бронирование Отелей |
| Тип | Консольное (CLI) |
| Цель | Разработка демонстрационного приложения для управления каталогом отелей, реализующего паттерн MVC и функционал |
| Паттерн проектирования | Model-View-Controller (MVC) |

2. Функциональные требования (Что должно делать приложение)

Приложение должно предоставлять пользователю следующие возможности через консольное меню:

1. Просмотр каталога:

- Отображение полного списка всех отелей, включая их основные характеристики (Название, Город, Рейтинг, Цена, Статус доступности).

2. Поиск отелей:

- Возможность поиска и отображения отелей по заданному Городу.

3. Бронирование (имитация):

- Возможность "забронировать" отель по его **точному Названию**.
- При успешном бронировании статус отеля должен измениться на "Забронирован" (available = False).

- Система должна выдавать соответствующее сообщение об успехе или ошибке (если отель не найден или уже забронирован).

4. Генерация HTML-отчёта (Представление **Jinja2**):

- Создание статического HTML-файла (catalog.html), который содержит полный, красиво отформатированный список всех отелей из текущего каталога.
- Для генерации должен использоваться шаблонный движок **Jinja2**.

5. Выход:

- Корректное завершение работы приложения.

3. Требования к архитектуре (Паттерн MVC)

Приложение должно быть строго разделено на три основные компонента, каждый из которых должен быть реализован в отдельном файле:

3.1. Модель (model.py)

- **Класс Hotel:** Хранение данных об одном отеле (Название, Город, Рейтинг, Цена, Доступность).
- **Класс HotelCatalog:** Управление коллекцией отелей (создание, хранение тестовых данных).
- **Бизнес-логика:** Содержит методы для:
 - Получения всех отелей (get_all_hotels).
 - Поиска отелей по городу (find_hotels_by_city).
 - Изменения статуса бронирования (book_hotel).

3.2. Представление (view.py и hotel_template.html)

- **Консольный вывод (ConsoleView):**
 - Отвечает исключительно за вывод информации (приветствие, меню, списки отелей) в консоль.
 - Отвечает за получение пользовательского ввода (выбор меню, название отеля, город для поиска).
 - Не содержит никакой бизнес-логики.
- **HTML-генерация (функция generate_html_catalog):**
 - Использует **Jinja2** для загрузки шаблона (hotel_template.html).
 - Принимает список объектов отелей и генерирует на их основе HTML-строку.
 - Записывает полученную HTML-строку в файл catalog.html.

3.3. Контроллер (controller.py и main.py)

- **Класс Controller:**
 - Связывает Модель и Представление.
 - Содержит основной цикл программы (run).
 - Обрабатывает выбор пользователя из меню.
 - Вызывает соответствующие методы Модели для работы с данными.
 - Вызывает соответствующие методы Представления для отображения результатов или получения ввода.

4. Требования к исходным данным

- На старте приложение должно иметь небольшой набор **тестовых данных** (минимум 4 отеля) внутри класса HotelCatalog для демонстрации функциональности.
- Данные должны быть представлены в виде экземпляров класса Hotel
-