

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

КУРСОВОЙ ПРОЕКТ
ЗАЩИЩЕН С ОЦЕНКОЙ
РУКОВОДИТЕЛЬ

Ст.преподаватель

Е.О. Шумова

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К КУРСОВОМУ ПРОЕКТУ

Разработка приложения для организации взаимодействия объектов при
заданных критериях

по дисциплине: ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.

Z1431

М.Д. Быстров

подпись, дата

инициалы, фамилия

Санкт-Петербург 2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное автономное образовательное учреждение высшего образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ И ПРОГРАММНОЙ ИНЖЕНЕРИИ

Задание
на курсовой проект по дисциплине
«Объектно-ориентированное программирование»

Студенту группы Z1431 Быстров М.Д.
№ группы Ф.И.О.

Тема «Разработка приложения для организации взаимодействия объектов при заданных критериях»

Исходные данные: Разработка иерархии классов для обеспечения работы гостиницы

Проект должен содержать:

- анализ предметной области
- разработку классов
- разработку тестового приложения
- оформление пояснительной записки по результатам выполнения проекта
- создание презентации к проекту

Срок сдачи законченного проекта _____

Руководитель проекта _____ ст.преп. Е.О.Шумова

Дата выдачи задания 01.09.2023 г.

Оглавление

Введение.....	5
1. Постановка задачи	6
1.1 Анализ предметной области.....	6
1.1.1 Основные сущности.....	6
1.1.2 Вспомогательные сущности.....	6
1.2 Формулировка технического задания	6
2. Проектирование классов	7
2.1 Классы сущностей.....	7
2.1.1 Класс «Room»	7
2.1.2 Класс «Bedroom»	8
2.1.3 Класс «Bed».....	8
2.1.4 Класс «Customer»	8
2.1.5 Класс «Accommodation»	9
2.1.6 Диаграмма классов.....	9
2.2 Классы хранения данных.....	9
2.2.1 Класс «BaseContext».....	9
2.2.2 Класс «HostelDbContext».....	11
2.2.3 Диаграмма классов.....	12
2.3 Управляющие классы.....	13
2.3.1 Класс «IRequirement»	13
2.3.2 Класс «CapacityRequirement».....	13
2.3.3 Класс «RoomTypeRequirement»	13
2.3.4 Класс «BedRequirement».....	13
2.3.5 Класс «BedroomRequirement»	14
2.3.6 Класс «BathroomRequirement»	14
2.3.7 Класс «FloorNumberRequirement»	15
2.3.8 Класс «AreaRequirement»	15
2.3.9 Класс «RequirementSet»	16
2.3.10 Класс «RequirementSetBuilder»	16
2.3.11 Класс «RequirementRoomProvider».....	16

2.3.12	Диаграмма классов.....	17
2.4	Интерфейсные классы.....	17
2.4.1	Класс «RoomForm»	17
2.4.2	Класс «CustomersForm»	19
2.4.3	Класс «CreateAccommodationForm»	19
2.4.4	Класс «EditEntityForm».....	20
2.4.5	Диаграмма классов.....	21
2.5	Используемые паттерны проектирования	21
2.5.1	Паттерн «Singleton»	21
2.5.2	Паттерн «Builder» («Строитель»)	22

Введение

Предметная область курсового проекта – обеспечение работы гостиницы. В ходе выполнения проекта должно быть спроектировано и реализовано приложение для учета и распределения по свободным номерам приезжих гостей.

Приложение должно отвечать следующим требованиям: работа с базой данных, графический интерфейс, использование концепции ООП и паттернов проектирования.

В разделах «Постановка задачи», «Проектирование классов» настоящей пояснительной записки содержатся: описание определенных на основе анализа предметной области сущностей, информация о выбранных технологиях и инструментах разработки, диаграммы разработанных иерархий классов, перечисление используемых паттернов проектирования.

Далее, в разделе «Разработка приложения» описаны детали реализации приложения, приводятся изображения, содержащие пользовательский интерфейс тестового приложения.

В разделе «Тестирование» продемонстрирована работа приложения с использованием различных наборов тестовых данных.

В Приложении 1 размещен полный исходный код реализованной программы.

1. Постановка задачи

1.1 Анализ предметной области

Предметной областью курсового проекта является работа гостиницы в части регистрации и заселения клиентов. Процесс регистрации начинается при первом получении информации о клиенте, далее регистрируется факт заселения и выселения клиента.

По итогам анализа предметной области выделены следующие основные сущности:

1.1.1 Основные сущности

- Клиент (постоялец, гость) гостиницы
- Номер гостиницы
- Спальня номера
- Кровать в спальне

1.1.2 Вспомогательные сущности

- Поставщик данных БД
- Требования к номеру
 - Требование к вместимости
 - Требование к спальным местам
 - Требование к количеству комнат
 - Требование к количеству ванных комнат
 - Требование к этажу номера
 - Требование к площади номера
 - Требования к классу номера
- Набор требований к номеру
- Строитель набора требований к номеру
- Поставщик набора номеров в соответствии с требованиями к номеру

1.2 Формулировка технического задания

Спроектировать и реализовать программу для учета гостей гостиницы и данных об их проживании.

Программа должна иметь графический пользовательский интерфейс (GUI), выполняться в операционной системе MS Windows.

Язык программирования – C#. Среда разработки – Visual Studio.

Технология для создания пользовательского графического интерфейса – Windows Forms.

2. Проектирование классов

2.1 Классы сущностей

2.1.1 Класс «Room»

Класс «Room» описывает номер – сущность предметной области. Каждый экземпляр класса описывает один номер гостиницы.

Описание полей:

- Id – уникальный идентификатор
- Number – числовой код номера, в соответствии с распределением номеров в гостинице
- Name – название номера
- Type – тип (класс) номера
- BathroomsCount – количество ванных комнат в номере
- Floor – номер этажа, на котором располагается номер
- Area – площадь номера (м²)

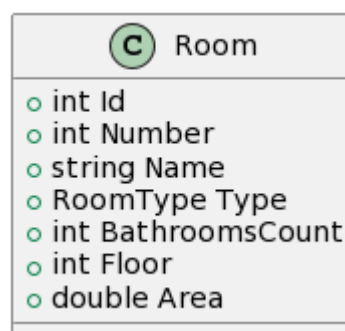


Рисунок 1 Класс Room

Все поля являются свойствами языка C#, т.е. имеют неявные методы доступа (getter, setter), инкапсулирующие поля. Далее во всех описаниях классов подразумевается, что доступ к данным, содержащимся в полях

экземпляров классов, происходит посредством неявных вызовов методов доступа.

2.1.2 Класс «Bedroom»

Класс «Bedroom» описывает спальню, принадлежащую номеру.

Описание полей:

- Id – уникальный идентификатор комнаты
- RoomId – идентификатор комнаты, которой принадлежит спальня
- Area – площадь спальни

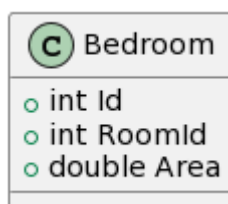


Рисунок 2 Класс Bedroom

2.1.3 Класс «Bed»

Класс «Bed» описывает кровать, расположенную в спальне номера.

Описание полей:

- Id – уникальный идентификатор кровати
- BedroomId – идентификатор спальни, в которой расположена кровать
- Capacity – вместимость кровати (количество человек)

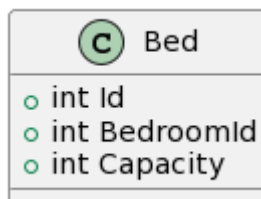


Рисунок 3 Класс Bed

2.1.4 Класс «Customer»

Класс «Customer» описывает гостя гостиницы (постоялец, заказчик).

Описание полей:

- Id – уникальный идентификатор гостя
- FullName – ФИО

- Birthday – дата рождения

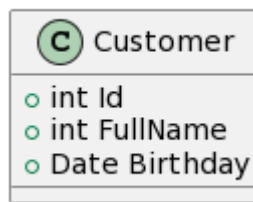


Рисунок 4 Класс Customer

2.1.5 Класс «Accommodation»

Класс «Accommodation» описывает заселение постояльца в номер.

Описание полей:

- Id – уникальный идентификатор заселения
- FromDate – дата заселения
- ToDate – дата выезда
- RoomId – ИД комнаты
- CustomerId – ИД постояльца

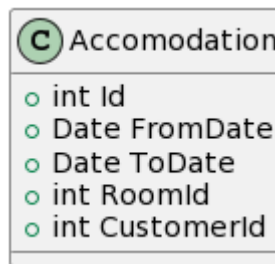


Рисунок 5 Класс Accommodation

2.1.6 Диаграмма классов

Поскольку сущности предметной области не связаны между собой отношениями, отражаемыми диаграммой классов, диаграмма классов для классов, описывающих сущности предметной области, приведена не будет.

2.2 Классы хранения данных

2.2.1 Класс «BaseContext»

Описание полей:

- _databaseFullFileName – путь к файлу базы данных

Описание методов:

- setDatabaseFullFileName – установить путь к файлу базы данных

- `getDatabaseFullFileName()` – получить путь к файлу базы данных
- `selectDatabaseFile()` – предоставить пользователю выбрать файл базы данных (либо имя нового файла для хранения данных)(статический)
- `getRooms` – получить номера
- `getRoom` – получить номер по идентификатору
- `addRoom` – добавить новый номер
- `updateRoom` – обновить существующий номер
- `deleteRoom` – удалить номер
- `getAccomodations` – получить все данные заселений
- `getAccomodation` – получить заселение по идентификатору
- `addAccomodation` – создать новое заселение
- `updateAccomodation` – обновить заселение
- `deleteAccomodation` – удалить заселение
- `getBedrooms` – получить спальни
- `getRoomBedrooms` – получить спальни по идентификатору номера
- `getBedroom` – получить спальню по идентификатору
- `addBedroom` – добавить спальню
- `updateBedroom` – обновить спальню
- `deleteBedroom` – удалить спальню
- `getBeds` – получить кровати
- `getBedroomBeds` – получить кровати по идентификатору спальни
- `getBed` – получить кровать по идентификатору
- `addBed` – добавить кровать
- `updateBed` – обновить кровать
- `deleteBed` – удалить кровать
- `getCustomers` – получить постояльцев
- `getCustomer` – получить постояльца по идентификатору

- addCustomer – добавить постояльца
- updateCustomer – обновить постояльца
- deleteCustomer – удалить постояльца
- clearDatabase – очистить базу данных

2.2.2 Класс «HostelDbContext»

Описание полей:

- _instance – экземпляр контекста (статическое)

Описание методов:

- getInstance – (статическое) – получить экземпляр контекста базы данных
- getVacantRooms – получить доступные комнаты на период дат
- createRoomAccommodation – создать заселение постояльца в комнату
- getRoomAccommodationOnDate – получить заселение в комнату на дату
- isRoomVacantOnDate – проверить, свободна ли комната на дату

В классе HostelDbContext реализован паттерн «Singleton». Назначение и устройство паттерна будет описано в разделе 2.5 «Используемые паттерны проектирования».

2.2.3 Диаграмма классов

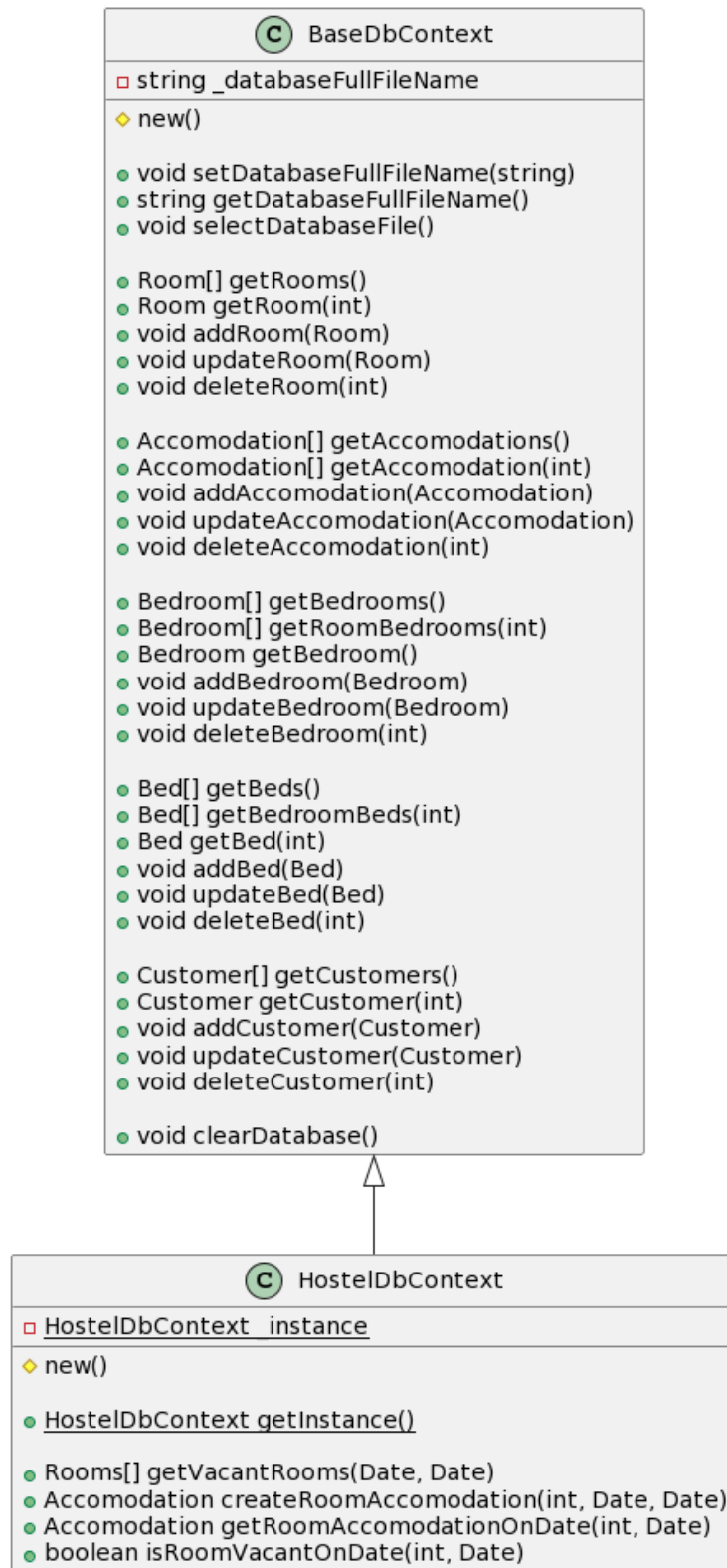


Рисунок 6 Диаграмма классов

2.3 Управляющие классы

2.3.1 Класс «IRequirement»

Класс «IRequirement» является интерфейсом для объектов, позволяющих проверить номер гостиницы на соответствие определенным требованиям.

Описание методов:

- checkRoom – проверка комнаты на соответствие требованиям

2.3.2 Класс «CapacityRequirement»

Класс «CapacityRequirement» предназначен для проверки комнаты на соответствие требованию по вместимости (кол-во человек).

Описание полей:

- minCapacity – минимально допустимая вместимость
- maxCapacity – максимально допустимая вместимость

Описание методов:

- checkRoom – проверка комнаты на соответствие требованиям
- getMinCapacity – получить минимальную вместимость
- setMinCapacity – задать минимальную вместимость
- getMaxCapacity – получить максимальную вместимость
- setMaxCapacity - задать максимальную вместимость

2.3.3 Класс «RoomTypeRequirement»

Класс «RoomTypeRequirement» - проверка номера на соответствие типа номера требуемому

Описание полей:

- roomTypes – массив допустимых типов номеров

Описание методов:

- checkRoom – проверка комнаты на соответствие требованиям
- getRoomTypes – получить массив типов номеров
- setRoomTypes – задать массив типов номеров

2.3.4 Класс «BedRequirement»

Класс «BedRequirement» - проверка наличия и вместимости кроватей

Описание полей:

- bedCount – количество кроватей
- bedCapacity – вместимость кроватей

Описание методов:

- checkRoom – проверка комнаты на соответствие требованиям
- getBedCount – получить кол-во кроватей
- setBedCount – установить кол-во кроватей
- getBedCapacity – получить вместимость кроватей
- setBedCapacity – задать вместимость кроватей

2.3.5 Класс «BedroomRequirement»

Класс «BedroomRequirement» - требование к наличию спален

Описание полей:

- minBedroomNumber – минимальное кол-во спален
- maxBedroomNumber – максимальное кол-во спален

Описание методов:

- checkRoom – проверка комнаты на соответствие требованиям
- getMaxBedroomNumber – получить максимальное кол-во спален
- setMaxBedroomNumber – задать максимальное кол-во спален
- getMinBedroomNumber – получить минимальное кол-во спален
- setMinBedroomNumber – задать минимальное кол-во спален

2.3.6 Класс «BathroomRequirement»

Класс «BathroomRequirement» - требование к кол-ву ванных комнат

Описание полей:

- minBathroomNumber – минимально допустимое кол-во ванных комнат
- maxBathroomNumber – максимально допустимое кол-во ванных комнат

Описание методов:

- checkRoom – проверка комнаты на соответствие требованиям

- getMaxBathroomNumber - получить минимально допустимое кол-во ванных комнат
- setMaxBathroomNumber - задать минимально допустимое кол-во ванных комнат
- getMinBathroomNumber - получить максимально допустимое кол-во ванных комнат
- setMinBathroomNumber - задать максимально допустимое кол-во ванных комнат

2.3.7 Класс «FloorNumberRequirement»

Класс «FloorNumberRequirement» - требование к этажу номера

Описание полей:

- minFloorNumber – минимально допустимый этаж
- maxFloorNumber – максимально допустимый этаж

Описание методов:

- checkRoom – проверка комнаты на соответствие требованиям
- getMinFloorNumber – получить минимально допустимый этаж
- setMinFloorNumber – задать минимально допустимый этаж
- getMaxFloorNumber – получить максимально допустимый этаж
- setMaxFloorNumber – задать максимально допустимый этаж

2.3.8 Класс «AreaRequirement»

Класс «AreaRequirement» - требование к площади номера

Описание полей:

- minArea – минимально допустимая площадь
- maxArea – максимально допустимая площадь

Описание методов:

- checkRoom – проверка комнаты на соответствие требованиям
- getMinArea – получить минимально допустимую площадь
- setMinArea – задать минимально допустимую площадь
- getMaxArea – получить максимально допустимую площадь

- setMaxArea – задать максимально допустимую площадь

2.3.9 Класс «RequirementSet»

Класс «RequirementSet» представляет из себя контейнер для множества экземпляров классов, реализующих интерфейс IRequirement. Назначение класса – проверить номер на соответствие нескольким требованиям.

Описание полей:

- requirements – массив требований

Описание методов:

- checkRoom – проверка комнаты на соответствие требованиям
- setRequirements – задать набор требований
- getRequirements – получить набор требований

2.3.10 Класс «RequirementSetBuilder»

Класс «RequirementSetBuilder» является классом, спроектированным с использованием паттерна «Строитель» (“Builder”). Экземпляр класса позволяет быстро сконструировать набор требований к номеру посредством последовательного вызова методов.

Описание методов:

- addCapacityRequirement – добавить требование к вместимости
- addRoomTypeRequirement – добавить требование к типу номера
- addBedRequirement – добавить требование к спальным местам
- addFloorNumberRequirement – добавить требование к этажу
- addAreaRequirement – добавить требование к площади
- addBedroomRequirement – добавить требование к спальням
- addBathroomRequirement – добавить требование к ванным комнатам
- buildRequirementSet – построить RequirementSet

2.3.11 Класс «RequirementRoomProvider»

Класс «RequirementRoomProvider» позволяет предоставить набор номеров в соответствии с заданными требованиями.

Описание полей:

- requirements – набор требований

Описание методов:

- getRooms – получить комнаты, соответствующие требованиям
- getVacantRooms – получить только свободные для заселения комнаты

2.3.12 Диаграмма классов

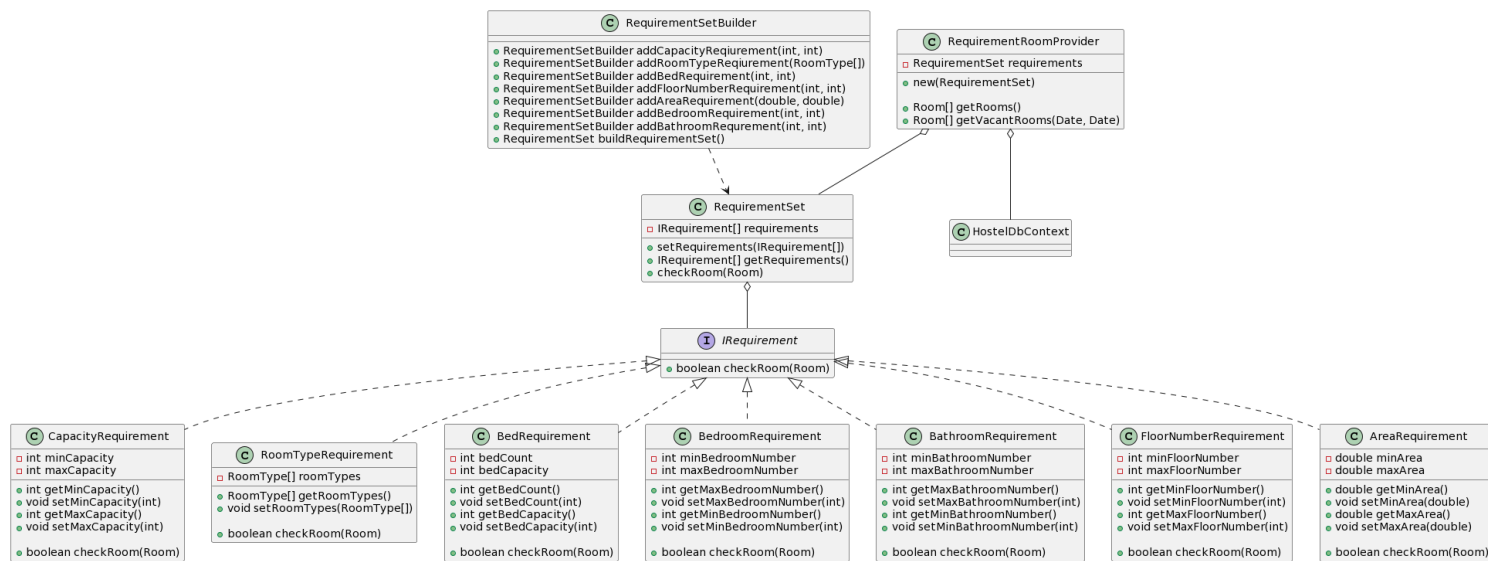


Рисунок 7 Диаграмма классов

2.4 Интерфейсные классы

2.4.1 Класс «RoomForm»

Класс «RoomForm» представляет из себя класс обработки пользовательского ввода на главной форме приложения.

Описание полей:

Первый тип полей – экранные фильтры для поиска номеров в соответствии с требованиями.

- RoomTypeField – выбор типа номера
- MinCapacityField – выбор минимальной вместимости номера
- MaxCapacityField – выбор максимальной вместимости номера
- OnePlaceBedNumberField – выбор кол-ва односпальных кроватей
- TwoPlaceBedNumberField – выбор кол-ва двухспальных кроватей

- MinBathroomNumberField – выбор минимального кол-ва ванных комнат
- MaxBathroomNumberField – выбор максимального кол-ва ванных комнат
- MinAreaNumberField – выбор минимальной площади номера
- MaxAreaNumberField – выбор максимальной площади номера
- MinFloorNumberField – выбор минимального этажа номера
- MaxFloorNumberField – выбор максимального этажа номера
- MinBedroomNumberField – выбор минимального кол-ва спален в номере
- MaxBedroomNumberField – выбор максимального кол-ва спален в номере

Также на форме присутствуют элементы отображения данных и пользовательского ввода:

- RoomGrid – сетка данных комнат
- BedroomGrid – сетка данных спален
- BedGrid – сетка данных кроватей
- AccomodationGrid – сетка данных заселений
- CreateRoomButton – кнопка «Создать номер»
- EditRoomButton – кнопка «Редактировать номер»
- DeleteRoomButton – кнопка «Удалить номер»
- CreateBedroomButton – кнопка «Создать спальню»
- EditBedroomButton – кнопка «Редактировать спальню»
- DeleteBedroomButton – кнопка «Удалить спальню»
- CreateBedButton – кнопка «Создать кровать»
- EditBedButton – кнопка «Редактировать кровать»
- DeleteBedButton – кнопка «Удалить кровать»
- CreateAccomodationButton – кнопка «Заселить гостя»
- DeleteAccomodationButton – кнопка «Отменить заселение»

- CustomersButton – кнопка «Гости»
- ApplyFiltersButton – кнопка «Применить фильтры»

Описание методов:

- getRequiredRooms – получить комнаты, соответствующие требованиям, указанным в экранном фильтре. Вызывается при нажатии на кнопку «Применить фильтры»

2.4.2 Класс «CustomersForm»

Класс «CustomersForm» содержит управляющие элементы формы «Гости», на которой доступно редактирование персональных данных гостей гостиницы.

Описание полей:

- CustomerGrid – сетка с данными гостей
- NameField – поле для ввода ФИО
- BirthdayField – поле для ввода даты рождения
- EditButton – кнопка «Редактировать»
- CreateButton – кнопка «Создать»
- RemoveButton – кнопка «Удалить»

2.4.3 Класс «CreateAccommodationForm»

Класс «CreateAccommodationForm» описывает элементы формы для заселения гостя.

Описание полей:

- FromDateField – поле начальной даты заселения
- ToDateField – поле конечной даты заселения
- CustomerIdField – поле идентификатора гостя
- CustomerNameField – поле ФИО гостя
- CreateAccommodationButton – кнопка «Создать заселение»
- CreateCustomerButton – кнопка «Гости» (переход к форме «Гости»)

Описание методов:

- createAccommodation – создание заселения. Используется экземпляр HostelDbContext

2.4.4 Класс «EditEntityForm»

Класс «EditEntityForm» представляет из себя класс формы для создания/редактирования сущностей БД пользователем (номера, спальни, кровати). Наполнение формы осуществляется динамически на основании полей сущности.

Описание полей:

- entity – при редактировании – данные редактируемой сущности
- SaveButton – кнопка «Сохранить»

Описание методов:

- saveEntity – сохранить/обновить сущность в БД. Используется HostelDbContext

2.4.5 Диаграмма классов

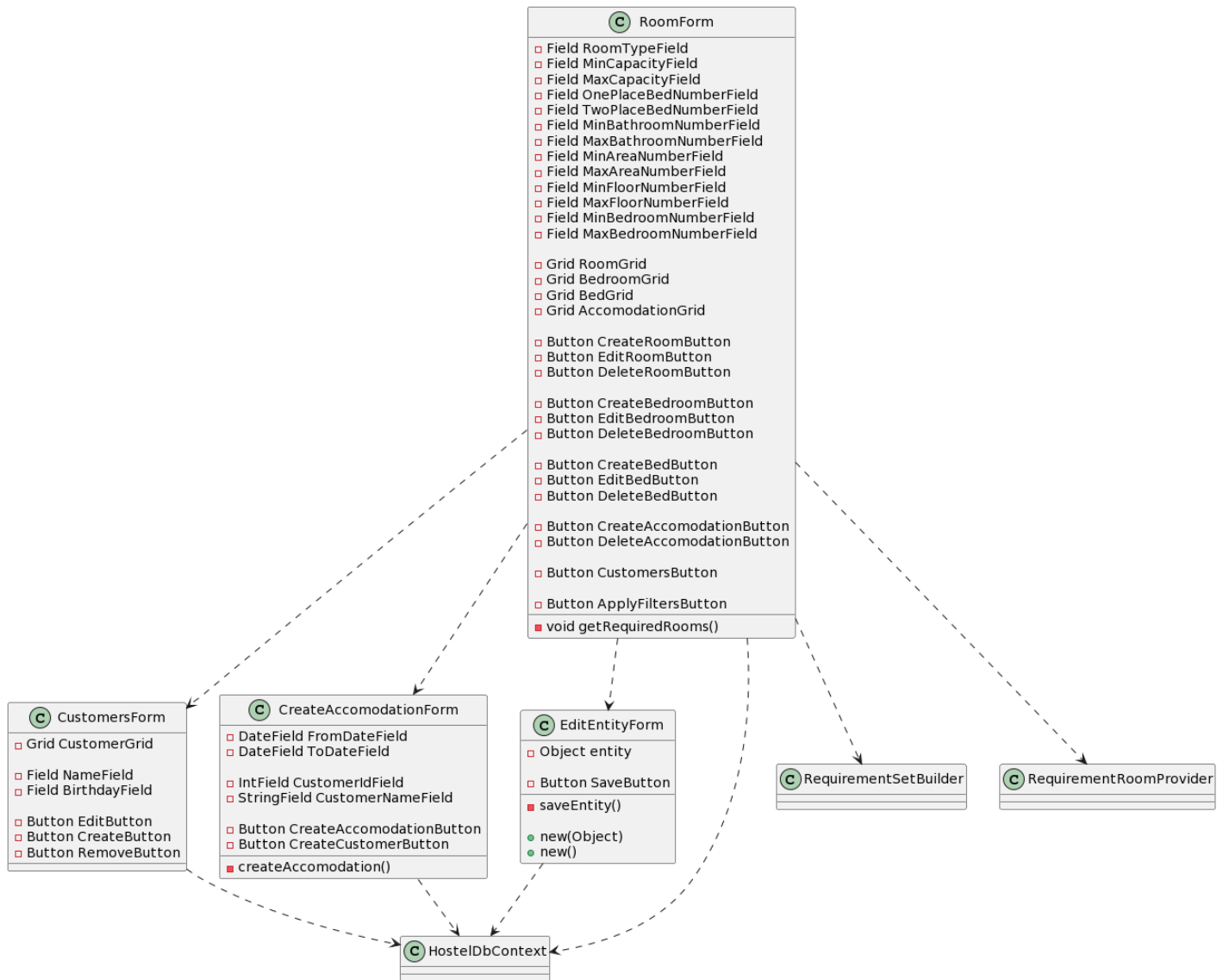


Рисунок 8 Диаграмма классов

2.5 Используемые паттерны проектирования

2.5.1 Паттерн «Singleton»

С помощью паттерна Singleton спроектирован класс `HostelDbContext`. Суть паттерна заключается в существовании только одного экземпляра класса на протяжении работы программы. Реализуется паттерн с помощью статических методов, полей и примитивов синхронизации потоков.

Использование паттерна позволяет получить доступ к объекту из любого места в приложении, в данном случае к поставщику данных из БД, что позволит избежать множественных конкурентных чтений/записей в файл БД, а также упростит архитектуру приложения.

2.5.2 Паттерн «Builder» («Строитель»)

Паттерн «Builder» позволяет упростить конструирование сложных объектов путем вызова соответствующих методов. Это позволяет избежать множественных вызовов методов конструируемого объекта (для предоставления зависимостей) после его создания, либо передачи большого количества переменных в конструктор, если класс конструируемого объекта спроектирован таким образом, чтобы невозможно было создать экземпляр без предоставления всех необходимых зависимостей. Класс, реализованный в соответствии с паттерном, упрощает конструирование. Как правило, каждый метод, предназначенный для настройки, возвращает указатель на сам экземпляр «Builder», что позволяет вовсе не создавать лишних локальных переменных.

Часто паттерн используется в тех случаях, когда необходимо конструировать схожие (но разные по типу, данным) объекты разным способом. В таком случае существует несколько «Builder»-классов, которые реализуют один интерфейс или унаследованы от одного базового класса. В данном курсовом проекте этот подход в использовании паттерна не применен.