



Московский государственный университет имени М.В. Ломоносова

Факультет Вычислительной математики и кибернетики

Кафедра Алгоритмических языков

Отчет по заданию практикума по объектно-ориентированному  
программированию

## **Система поддержки бронирования и заселения гостиницы**

**Выполнил:**

студент 424 группы

Сеник Алексей Михайлович

Москва, 2023

## Оглавление

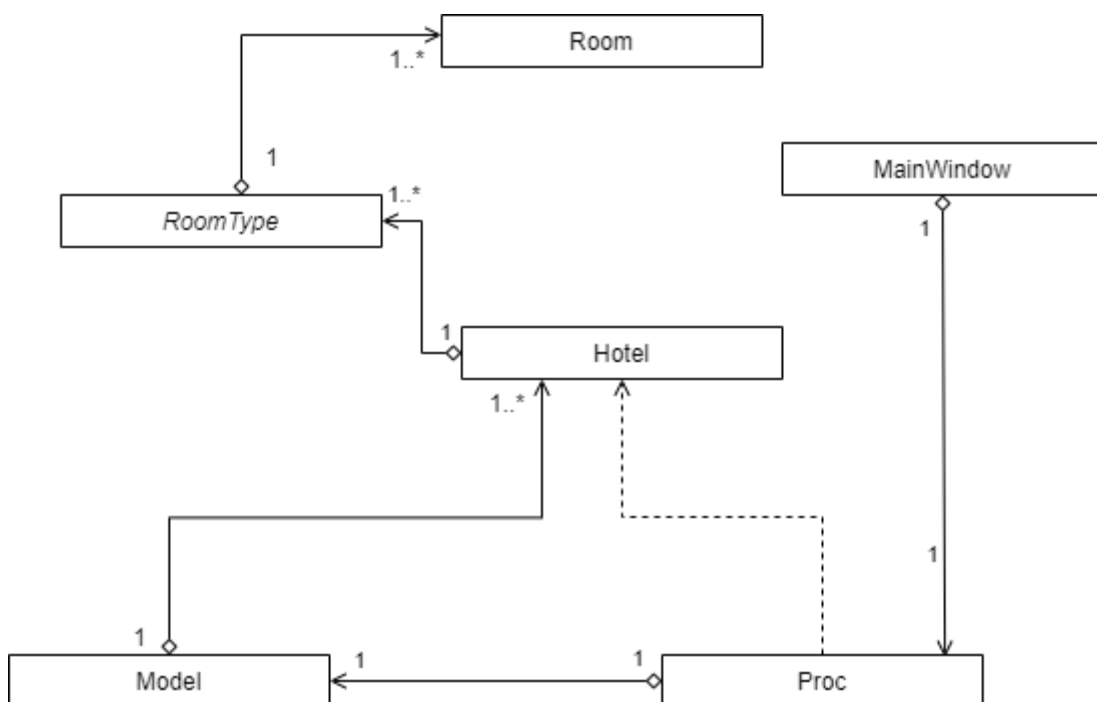
Уточнение постановки задачи .....	3
Диаграмма классов.....	3
Текстовые спецификации интерфейса .....	4
Диаграмма объектов .....	6
Инструментальные средства.....	7
Описание файловой структуры программы.....	7
Пользовательский интерфейс .....	8

## Уточнение постановки задачи

Задача – создать компьютерную систему, автоматизирующую управление занятостью номеров гостиницы. Система должна:

- обрабатывать входной поток заявок, бронирующих определенные типы номеров на определенный срок, и заявок на заселение в текущий момент;
- хранить информацию о фактической занятости всех номеров и о их занятости в ближайшие дни (учитываются уже оплаченные вперед дни), а также сведения о произведенной брони номеров;
- при нехватке нужных номеров использовать пустующие номера большей комфортности по меньшей цене;
- моделировать входной поток заявок на бронирование и поселение (вид и параметры каждой заявки определяются случайным образом);
- выводить статистику заселения номеров, выполненных заявок, процент загруженности отдельных категорий номеров и гостиницы в целом.

## Диаграмма классов



## Текстовые спецификации интерфейса

```
//класс, описывающий комнату
class Room {
public:
    // свободна ли комната
    bool getVacant() const;

    // кол-во бронирований
    bool getNumOfBooks() const;

    // обновление списка бронирований
    void update(int day);

    // обработка заявки
    bool processRequest(bool isBooking, std::pair<int, int> &date);
};

// класс, описывающий совокупность номеров одного типа
class RoomType {
public:
    // цена комнат данного типа
    int getCost() const;

    // кол-во комнат
    int getNumber() const;

    // кол-во бронирований всех комнат данного типа
    int getBookings() const;

    // кол-во занятых в данный момент комнат
    int getOccupied() const;

    // кол-во заявок на комнаты данного типа за весь период
    int getOccupiedAllTime() const;

    // процент загрузки
    double getOccupancyPercentage() const;

    // комнаты
    std::vector<Room> getRooms() const;

    // конструктор
    RoomType(int typeCost, int typeNumber);

    // обновление всех комнат данного типа
    void update(int day);

    // обработка заявки
    bool processRequest(bool isBooking, std::pair<int, int> &date);
};

// класс, описывающий всю гостиницу
```

```

class Hotel {
public:
    // конструктор
    Hotel(std::vector<int> &cnt, std::vector<int> &prices);

    // обновление всех комнат в отеле
    void update(int day);

    // обработка заявок
    bool processRequest(int type, bool isBooking, std::pair<int, int>
&dates);

    // комнаты всех типов
    std::vector<RoomType> getRooms() const;

    // кол-во заявок
    int getTotalRequests() const;

    // увеличение кол-ва заявок
    void incTotalRequests();

    // кол-во успешных заявок
    int getDoneRequests() const;

    // увеличение кол-ва успешных заявок
    void incDoneRequests();

    // прибыль
    int getProfit() const;

    // увеличение прибыли
    void incProfit(int income);
};

```

```

// класс, отвечающий за моделирование работы системы
class Model {
public:
    // период моделирования
    int getPeriod() const;

    // кол-во заявок
    int getTotalRequests() const;

    // кол-во успешных заявок
    int getDoneRequests() const;

    // прибыль
    int getProfit() const;

    // текущий день
    int getDay() const;

    // кол-во занятых на данный момент комнат всех типов
    std::vector<int> getOccupied() const;

    // процент загрузки номеров всех типов
    std::vector<double> getOccupancyPercentage() const;

```

```

// кол-во броней номеров всех типов
std::vector<int> getBookings() const;

// кол-во комнат каждого типа
int getNumberOfRooms(int type);

// конструктор
Model(int per, Hotel &hot);

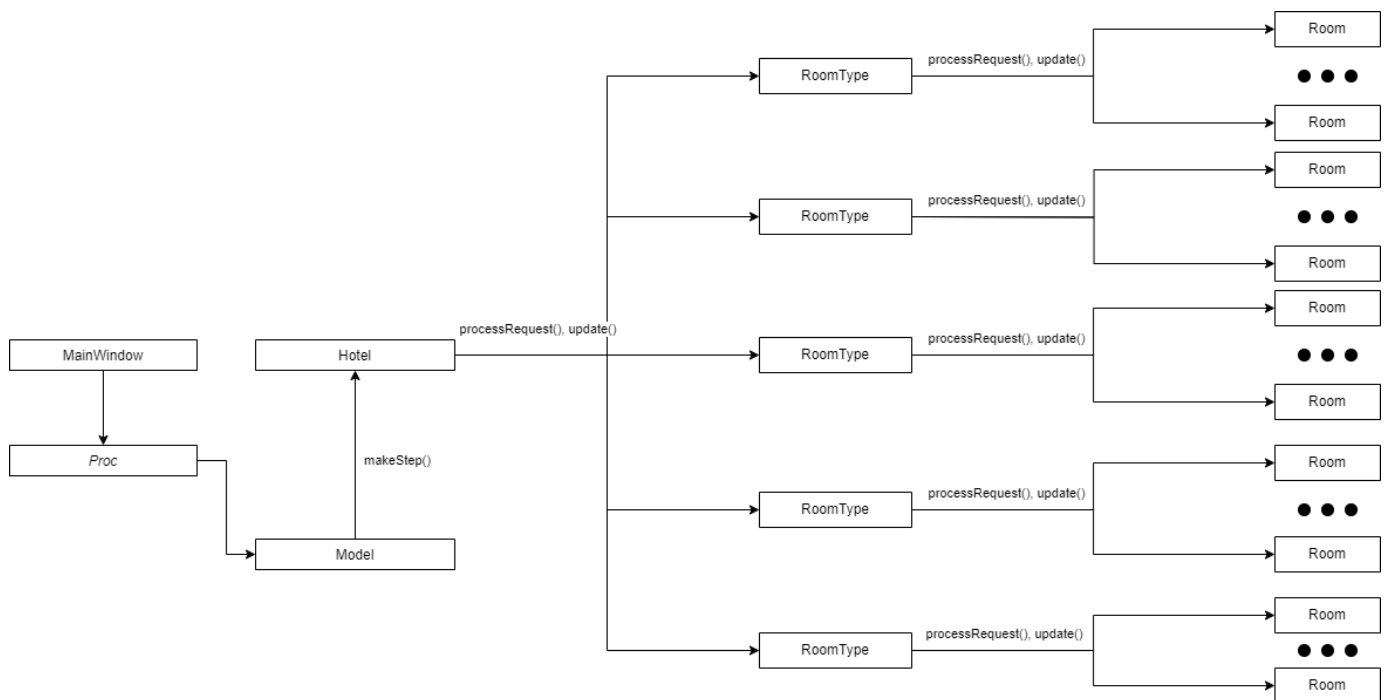
// шаг моделирования
void makeStep();
};

// окно для ввода параметров (без явного публичного интерфейса)
class MainWindow {};

// окно, отображающее моделирование (без явного публичного интерфейса)
class Proc {};

```

## Диаграмма объектов



## Инструментальные средства

- Язык программирования – C++;
- Среда разработки – Qt Creator;
- Используемая графическая библиотека – Qt.

## Описание файловой структуры программы

Корневой каталог:

- Headers:
  - Mainwindow.h
  - Proc.h
  - Hotel.h
  - Model.h
  - Room.h
  - Roomtype.h
- Sources:
  - Main.cpp
  - Mainwindow.cpp
  - Proc.cpp
- Forms:
  - Mainwindow.ui
  - Proc.ui

## Пользовательский интерфейс

Сначала перед пользователем появляется окно с вводом параметров, в котором он может ввести период моделирования (от 12 до 30 дней) и суммарное кол-во номеров (от 20 до 30). Далее необходимо ввести кол-во и цену каждого типа номера и нажать кнопку «Начать моделирование». Если суммарное кол-во номеров не совпадает с суммой чисел номеров каждого типа, то появляется окно с ошибкой.

Характеристики номеров		
	Кол-во номеров	Цена, Р
Одноместные	6	1000
Двуместные (с диваном)	5	2000
Двуместные	4	3000
Полулюкс	3	4000
Люкс	2	5000

Начать моделирование

Если все данные были введены правильно, то появляется окно с информацией о занятости номеров в текущий день и общей статистикой. Чтобы перейти на следующий день, необходимо нажать соответствующую кнопку в окне.



Управление гостиницей					
День 5 из 12					
	Одноместный	Двуместный (с диваном)	Двуместный	Полулюкс	Люкс
Цена	1000 Р	2000 Р	3000 Р	4000 Р	5000 Р
Занято номеров	1	1	1	0	1
Свободно номеров	5	4	3	3	1
Кол-во броней	4	2	2	2	1
Загруженность номеров	6%	12%	5%	13%	10%
Общая статистика					
Всего заявок	28				
Выполнено заявок	24				
Прибыль	187000 Р				
			Следующий день		

После окончания моделирования будет отображена окончательная статистика и информация по номерам. При это будет выведено специальное сообщение в верхней части окна, а кнопка «Следующий день» станет неактивной.

Управление гостиницей					
Моделирование завершено!					
	Одноместный	Двуместный (с диваном)	Двуместный	Полулюкс	Люкс
Цена	1000 Р	2000 Р	3000 Р	4000 Р	5000 Р
Занято номеров	0	0	0	0	0
Свободно номеров	6	5	4	3	2
Кол-во броней	0	0	0	0	0
Загруженность номеров	29%	52%	40%	46%	46%
Общая статистика					
Всего заявок	81				
Выполнено заявок	54				
Прибыль	317200 Р				
			Следующий день		