1.Увод

Информационна система Хотел е една от най-известните системи при обработка на данни за Хотели по целия свят.

Основните цели и задачи са, да се пресъздаде максимално много една реалистична информационна система и да се удовлетворят колкото се може по-добре поставените изисквания.

1. От страна на потребителя :

удобен интерфейс, включващ навигационно меню(чрез въвеждане само на низ – команда, от клавиатурата за различните функции, които може да избере);

2) От страна на разработчика:

2.1) диаграма, съдържаща нагледно йерархията, използвана за направата на проекта;

2.2) изчистен и четим код;

3) Структурата на документацията: включва увод, преглед на предметната област, проектиране, реализация, заключение, връзка към GitHub,

и използвана литература.

2. Преглед на предметната област

2.1) Основните концепции, които са използвани в проекта, са свързани с основните принципи на Обектно ориентираното програмиране. Използвани са “гетъри”, “сетъри“, дефолтни и параметрични конструктори, вектор и стринг, предефинирани оператори на местата, на които е нужно и други ООП парадигми.

Кодът е разделен на отделни класове, в зависимост от изискванията за реализирането на функциите.

2.2) Сложността на зададената задача се състои в това да се реализират всички зададени операции.

2.3) Подходът за разрешаването на проблемите е стандартното представяне на една информационна система на Хотел.

2.4) Потребителят трябва да въведе името, номера и адреса си. Има право да избира какво иска да направи (регистрация в даден период от дни, освобождаване на стая, искане на справка за периода, в който е използвал стаята, опция за настаняване на даден брой гости и др.

3.Проектиране

Архитектурата на проекта е следната:

1. Имаме клас Хотел, в който са декларирани функциите от командния ред (open, close, save, save as, help) и функциите, които се изискват да се поддържат от системата (report, availability, checkout and etc.)
2. Имаме клас Стая, който съдържа информация какво е нужно за създаването на една стая в Хотела.
3. Клас Дата, който представя подадена дата от потребителя във формата ([ISO 8601](https://bg.wikipedia.org/wiki/ISO_8601)- година | месец | ден).
4. Клас Клиент, съдържащ информация какво е нужно за регистрацията на клиент в Хотел, подавайки име, телефонен номер и адрес.
5. Мейн функция - в нея се извикват функциите, дефинирани в клас Хотел. Тук потребителят може да избере команда, (името на някоя от декларираните функции в Хотел), с която да продължи.

3.1) Обща архитектура

3.2) Диаграми на класовете

4. Реализация

Реализация на (public) методите на клас Стая:

Room(); - подразбиращ се конструктор, чрез който се създават n на брой стаи;

Room(const Date& from, const Date& to, const std::string note);

Room(const Date& from, const Date& to, const std::string note,size\_t guests);

- два Overloaded конструктори с цел опционално избиране от страна на потребителя дали да въведе стойност за променливата guests или пък не;

size\_t getRoomNumber() const;

* Функции “гетъри” с цел достъп до член - данните от private секцията на класа. Гетърите притежават const в края на декларацията си, затова те не могат да променят член – данните на класа, а само извличат информация за тях.

Date& getDateTo() ;

Date& getDateFrom() ;

size\_t getAvailableBeds() const;

string getNote() const;

size\_t getGuests() const;

void clearRoom(); - функция, която изчиства данните за дадена стая след освобождаването й.

bool isAvailable(); - функция, която проверява дали търсената стая е свободна или не.

void setNote(string note);

* Функции “сетъри“ (мутатори), променят член-данните на класа;

void setIsAvailable(bool isAvailable);

void setDateFrom(const Date& from);

void setDateTo(const Date& to); void setRoomNumber(size\_t rno); void setGuests(size\_t guests);

void setAvailableBeds(size\_t availableBeds);

friend std::istream& operator>>(std::istream& in, Room& room);

friend std::ostream& operator<<(std::ostream& out, const Room& room);

* Предефинирани са двата оператори за извличане и вмъкване, но тъй като първият им операнд е поток, предефинирането им като член-функции на класа е невъзможно. Обаче те трябва да имат пряк достъп до компонентите на класа, затова ги предефинирам като функции-приятели на класа.

Реализация на (public) методите на клас Хотел:

Hotel(size\_t); - дефолтен конструктор, в който се извиква функцията за инициализиране на броя на стаите ;

void checkIn(const Date& from, const Date& to, const string note);

void checkIn(const Date& from, const Date& to, const string note,size\_t guests); - две предефинирани функции с цел опционално избиране от страна на потребителя дали да въведе стойност за променливата guests или пък не;

void availability(Date& date);- извежда списък на свободните стаи на (дата);

void checkOut(size\_t);- освобождаване на заета стая с номер;

void report(const Date& from, const Date& to);- извежда справка за използването на стаи в даден период, също така извежда и броя на дните, в които е била използвана;

Room\* find(size\_t beds, const Date& from, const Date& to);- намира подходяща свободна стая с поне <beds> на брой легла в период въведен от потребителя, като при наличие на повече свободни стаи се предпочитат такива с по-малко на брой легла;

void unavailable(size\_t roomNum,const Date& from, const Date& to,string note);-обявява стая с номер в даден период за временно недостъпна и се добавя бележка;

void open();

* Функции за командния ред

void close();

void save();

void saveAs();

void help();

size\_t getRoomsCount(); - спомагателна функция, за да проверя дали правилно се инициализират стаите;

size\_t getDaysFromDate(const Date&); - дава броя на дните от 0-левата година до сегашния ден;

size\_t getDaysBetweenDate(const Date&, const Date&); -на тази функция се подават две дати и използвайки, size\_t getDaysFromDate(const Date&),ги конвертира в дни и ги изважда една от друга. И модула на това число са броят на дните между двете дати;

void initializeRooms(size\_t);- задава броя на стаите;

void freeUpRooms(); - изчиства стаите и се използва в командата exit;

Реализация на (public) методите на клас Дата:

Притежава един подразбиращ се конструктор и един с параметри. Има член – функции за достъп, мутатори, предефинирани поточни оператори (<<, >>) и в допълнение:

void Print() const; -функция, която принтира датата във вид (година|месец|ден)

friend bool operator>(const Date&, const Date&);

Три предефинирани оператора за „“>“ , „<“ и оператор „==“;

friend bool operator<(const Date&, const Date&);

bool operator==(const Date& p) const;

Реализация на (public) методите на клас Клиент:

Класът е предназначен да се създава обект Клиент, като се въвеждат име, тел. номер и адрес на клиентите.

Притежава един подразбиращ се конструктор и деструктор. Има член – функции за достъп, , мутатор и в допълнение функцията:

bool isValidPhone(string \_phone);- проверява дали правилно се въвеждат телефонните номера на клиентите на хотела, като изискването е да започва с +, защото държавата е България и числата да са между 0-9 ;

Реализация на Main функцията:

int main(); - в main функцията се изпълняват всички останали функции

5.Заключение

Проектът има накъде да се усъвършенства и развива, като идеи за развитието му са: потребителят да може да избира стая с определена гледка и различни удобства, да плаща в брой или с карта при регистрацията си, да има определен брой заявки , които да могат да се обработват за ден и др.

Връзка към GitHub:

Използвана литература

[en.cppreference.com/w/](https://en.cppreference.com/w/)

[www.cppbuzz.com/](http://www.cppbuzz.com/)

Магдалина Тодорова *„Програмиране на C++“;*

[stackexchange.com/](https://stackexchange.com/)

[stackoverflow.com/](https://stackoverflow.com/)

[www.tutorialspoint.com/cplusplus/cpp\_templates.htm](https://www.tutorialspoint.com/cplusplus/cpp_templates.htm)

[www.w3resource.com/index.php](https://www.w3resource.com/index.php)