Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет   
информатики и радиоэлектроники»

Факультет инженерно-экономический

Кафедра экономической информатики

Дисциплина «Технологии разработки программного обеспечения»

|  |  |
| --- | --- |
|  | «К ЗАЩИТЕ ДОПУСТИТЬ» |
|  | Руководитель курсового проекта  Старший преподаватель кафедры  ЭИ  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.Н. Салапура |
|  | \_\_\_.\_\_\_\_.2023 |

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе

на тему:

**«Разработка Автоматизированной системы учета деятельности косметического салона»**

БГУИР КР 1-40 05 01-02 024 ПЗ

|  |  |
| --- | --- |
|  | Выполнил студент группы 272303 Тарабеш Аким Константинович  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |
|  | Курсовая работа представлена на проверку \_\_\_.\_\_\_\_.2023  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись студента) |

Минск 2023

**РЕФЕРАТ**

**Тарабеш, А.К.** Разработка автоматизированной системы учета деятельности косметического салона / А.К.Тарабеш. – Минск: БГУИР, 2022. – 64 с.

Пояснительная записка 64 с., 57 рис., 6 табл., 6 источников, 4 приложения

*программы учета деятельности косметического салона*, модели *UML*, *IDEF*0, *BPMN*, схемы алгоритмов, программное средство

*Цель* *курсовой работы*: минимизация времени сбора данных за счет автоматизации процесса заказа услуг.

*Методология проведения работы*: в процессе решения поставленных задач использованы принципы системного подхода, аналитические методы, методы компьютерной обработки экспериментальных данных и компьютерного моделирования.

*Результаты работы*: выполнена постановка задачи и определены основные методы ее решения; в ходе объектного моделирования системы построен ряд *UML*-диаграмм; разработаны модели бизнес-процессов предметной области на основе нотаций IDEF0 и BPMN; описаны основные алгоритмы работы программного средства; разработано руководство пользователя; выполнено тестирование программного средства, показавшее его соответствие функциональным требованиям, поставленным в задании на разработку.

Программный продукт разработан на языке *C* с применением *MS Visual Studio 2019*.

*Область применения результатов*: разработанное программное средство может применяться в кинотеатрах с целью автоматизации процесса бронирования билетов. Анализ данных может помочь пользователю при бронировании билета.

Разработанное программное средство полностью отвечает всем функциональным требованиям, необходимым при бронировании билетов в кинотеатре.

**СОДЕРЖАНИЕ**

[1 Анализ автоматизированной системы учета деятельности косметического салона 5](#_Toc153705875)

[1.1 Описание автоматизации учета деятельности косметического салона 5](#_Toc153705876)

[1.2 Обзор программ-аналогов по данной предметной области 5](#_Toc153705877)

[1.2.1 Салон красоты “Мильфей” 6](#_Toc153705881)

[1.2.2 Салон красоты “Манго” 7](#_Toc153705882)

[1.3 Методы и алгоритмы решения поставленной задачи 7](#_Toc153705883)

[2 Функциональное моделирование на основе стандарта idef0 9](#_Toc153705884)

[3 структура используемых данных 11](#_Toc153705885)

[4 описание созданных программных конструкций 17](#_Toc153705886)

[5 разработка и описание диаграммы классов приложения 22](#_Toc153705887)

[6 разработка и описание диаграммы вариантов использования приложения](#_Toc153705888)

[23](#_Toc153705888)

[7 блок-схема алгоритма всей программы и двух основных методов 24](#_Toc153705889)

[7.1 Блок-схема алгоритма SignIn 24](#_Toc153705890)

[8 описание алгоритма запуска приложения, его использование, результаты работы программы, тестирование обработки ошибок 25](#_Toc153705891)

ВВЕДЕНИЕ

Современное развитие индустрии красоты и ухода за собой неразрывно связано с прогрессивными технологиями, направленные на повышение качества обслуживания клиентов и автоматизацию процессов и задач.

Использования информационных технологий в сфере услуг обеспечивает функционирование косметических салонов на высшем уровне в следствии того, будет снижена нагрузка на руководящие должности. Потенциальный клиент сможет не стоять в очередях, а воспользовавшись системой эффективно осуществить поиск необходимой информации и оформления услуг.

Целью данного курсового проекта является разработка автоматизированной системы учета деятельности косметического салона.

Целесообразность работы обусловлена следующим фактором. В косметических салонах любой работник перегружен в связи с большим количеством клиентов, оформления отчетов и консультации посетителей. В связи с этим необходимым является использование в салоне автоматизированной системы, которая предоставляет необходимую информацию, обладает возможностью записи на любую интересующую услугу и эффективным учетом деятельности отдельным звеньев салона.

Салонный бизнес является достаточно новым видом деятельности, который в настоящее время активно развивается.

Таким образом, можно сделать вывод, что тема курсового проекта является достаточно актуальной в текущее время.

Задачи работы:

* ознакомиться с предметной областью;
* проанализировать процессы;
* разработать пояснительную записку и код.

Объектом исследования данной курсовой работы является косметический салон и его процессы, которые подлежат автоматизации.

# **Анализ автоматизированной системы учета деятельности косметического салона**

## **Описание автоматизации учета деятельности косметического салона**

Значения салона красоты в современном мире невозможно отнести к простому предоставлению косметических процедур. В современном обществе салоны красоты играют важную роль, предоставляя клиентам не только услуги по уходу за внешностью, но и различные процедуры для улучшения здоровья.

Услуги предлагаемые салонами можно разделить на 2 группы:

* меджевые услуги (парикмахерские, маникюр, педикюр)
* косметологические услуги (уход за кожей и здоровьем)

Данные услуги всегда принадлежат к числу более востребованных.

## **Обзор программ-аналогов по данной предметной области**

Для реализации актуального и конкурентоспособного продукта необходимо также исследовать приложения конкурентов.

В настоящее время существует множество специализированных программ автоматизации деятельности салона красоты, которые предназначены непосредственно для клиента и работников. Они обладают рядом достоинств:

* широкий спектр услуг;
* консультация по интересующей услуге;
* возможность просматривать отзывы о салоне;
* обратная связь.

Несмотря на ряд достоинств и большое количество возможностей, в большинство программ также имеет ряд недостатков:

* отсутствие конкретного прайс-листа;
* при записи на услугу, нет возможности записи к определенному специалисту.



## **Салон красоты “Мильфей”**

Салон красоты “Мильфей” предоставляет большой спектр косметических услуг (рисунок 1.2.1.1).



Рисунок 1.2.1.1 – Салон красоты “Мильфей”

Реализация заказа косметической услуги в салоне красоты “Мильфей” представлена следующим образом. Первоначально необходимо выбрать специалиста, дату и время и услугу (рисунок 1.2.1.2).

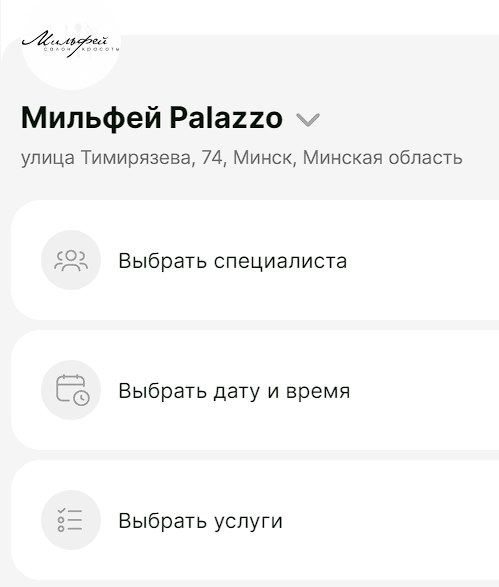


Рисунок 1.2.1.2 – Заказ услуги в салоне красоты “Мильфей”

Данный продукт дает возможность при записи на услугу не только выбор самой услуги, а также выбор интересующего специалиста и даты оказания услуги, что делает данный продукт весьма удобным в использовании.

## **Салон красоты “Манго”**

В салоне красоты “Манго” заказ реализован следующим образом. Для начала необходимо оставить заявку на услуги, после чего приходит ответ (рисунок 1.2.2.1).

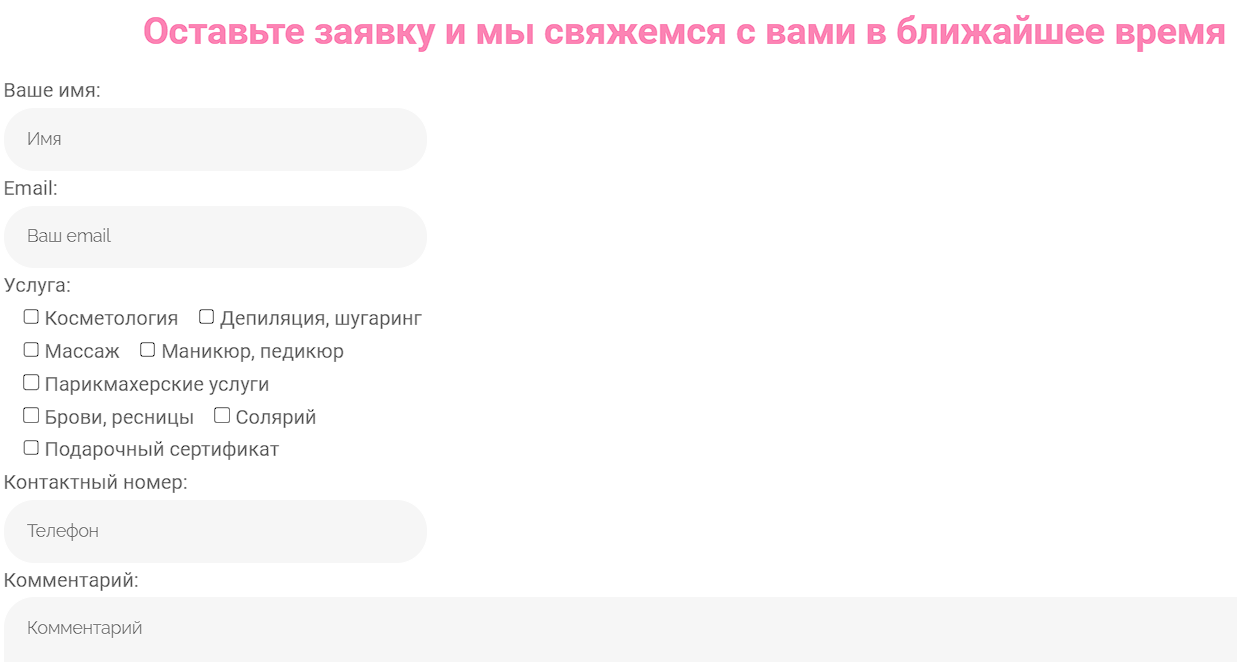


Рисунок 1.2.2.1 – Заказ услуги в салоне красоты “Манго”

Исходя из анализа данного продукта, можно сказать что он не удобен в использовании. Необходимо оставлять заявку на оказании услуги и ожидать обратной связи, что не всегда является желаемым для пользователя.

## **Методы и алгоритмы решения поставленной задачи**

Для автоматизации системы учета деятельности косметического салона можно использовать следующие методы и алгоритмы:

* создание удобного и интуитивно понятного интерфейса для заказа косметических услуг, который бы обеспечивал простоту использования для различных категорий пользователей;
* разработка алгоритма для расчета стоимости заказа и общей стоимости всех заказов;
* создание и управление файлами с данными, которые хранят информацию о заказах, услугах и данных пользователя;
* использование быстрых алгоритмов сортировок, поиска, фильтрации для уменьшения времени отклика программы;
* разработка архитектуры системы, определение сущностей системы и их взаимодействия между собой.

Таким образом были рассмотрены различные алгоритмы и методы для решения поставленной задачи.

# **Функциональное моделирование на основе стандарта IDEF0**

IDEF0 – нотация графического моделирования, используемая для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции. К её особенностям можно отнести:

* использование контекстной диаграммы;
* поддерживает декомпозиции;
* доминирование;
* выделение четырех типов стрелок.

На рисунке 2.1 представлена функциональная модель IDEF0 процесса “Предоставить косметическую услугу”, которая отражает общий принцип работы салона красоты.

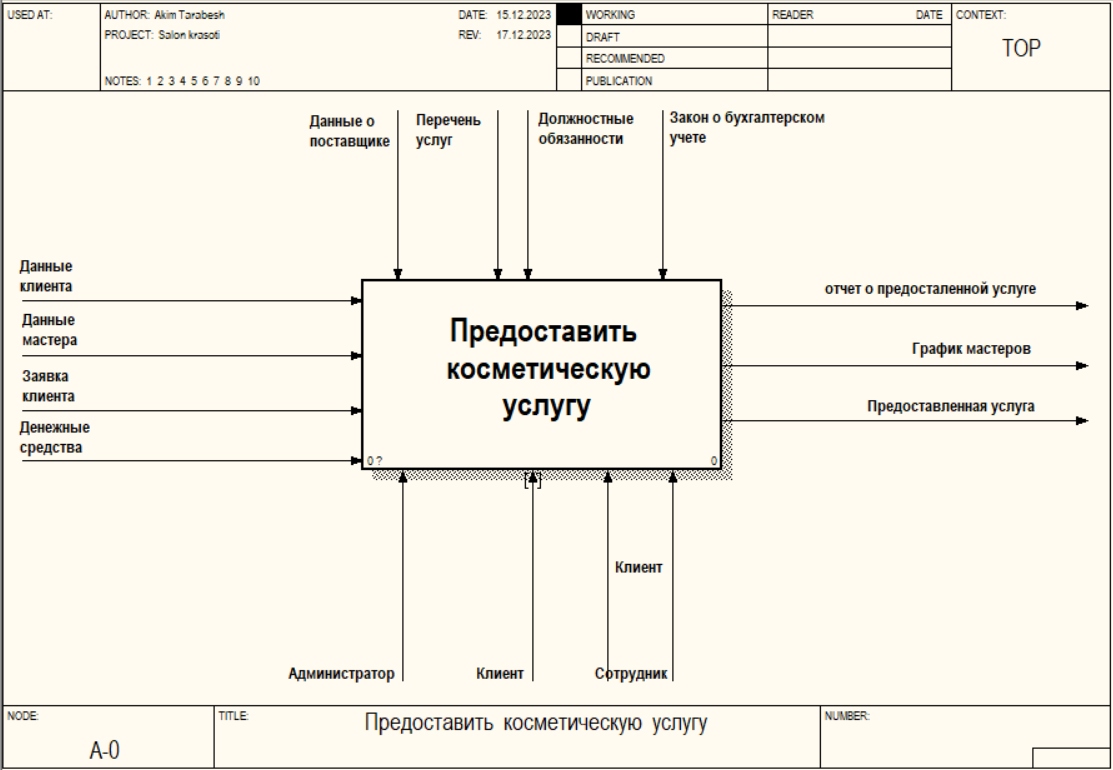


Рисунок 2.1 – Функциональная модель IDEF0 процесса “Предоставить косметическую услугу”

Для предоставления услуг требуется заявка от клиента на предоставление определенной косметической процедуры и его данные, данные о мастере, а также денежные средства. Ресурсами данного процесса являются сотрудник, администратор и клиент. В управлении выступают данные о поставщике, перечень услуг, должностные обязанности, закон о бухгалтерском учете. В результате выполнения процесса получим график мастеров, отчет о предоставленной услуге, предоставленная услуга.

Для более подробного анализа, рассмотрим декомпозицию функционального блока “предоставить косметическую услугу” (рисунок 2.2)

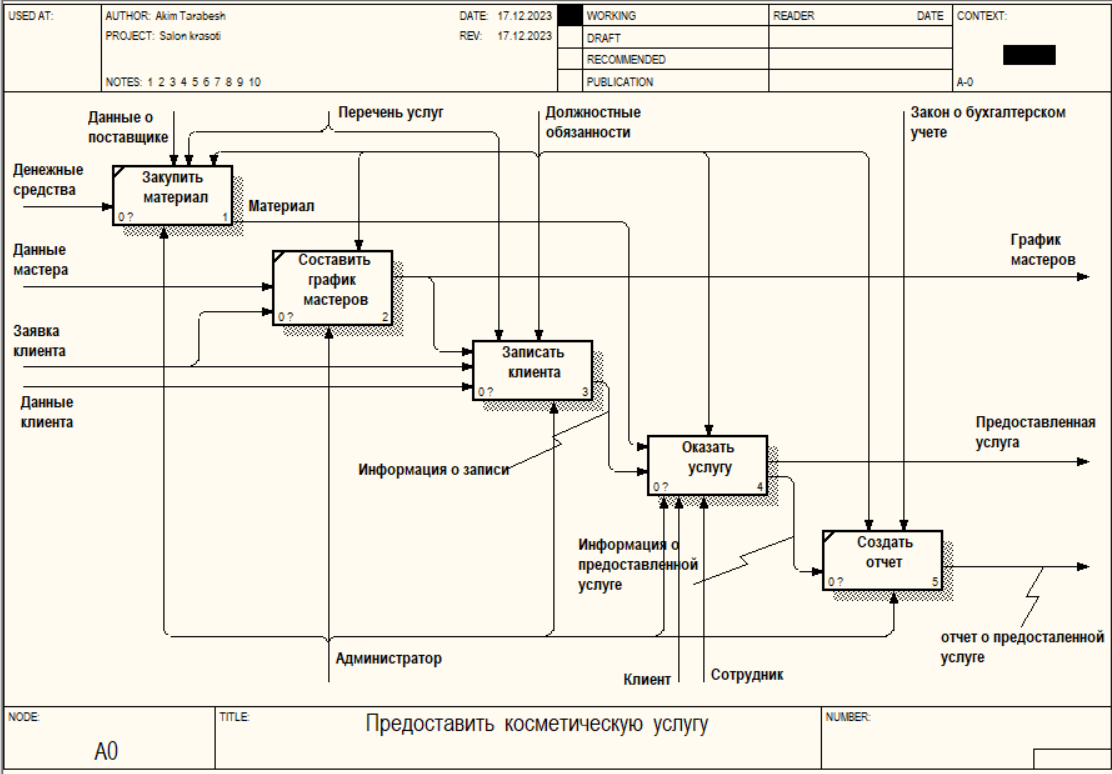


Рисунок 2.2 – Декомпозиция А-0 процесса “Предоставить косметическую услугу”

Модель отражает внутреннюю структуру процесса “Предоставить косметическую услугу”, исходя из которой можно увидеть, что она состоит из следующих подпроцессов: закупить материал, составить график мастеров, записать клиента, оказать услугу и создать отчет.

# **структура используемых данных**

Для организации данных были выбраны классы. Таблице приведенные ниже демонстрируют реализованные классы.

Таблица 3.1 – Переменные класса Account

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип данных | Назначение переменной |
| login | string | Хранит логин пользователя системы |
| password | string | Хранит пароль пользователя системы |
| role | string | Хранит роль пользователя системы |

Таблица 3.2 – Переменные класса Person

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип данных | Назначение переменной |
| name | string | Хранит имя пользователя системы |
| surname | string | Хранит фамилию пользователя системы |
| fname | string | Хранит отчество пользователя системы |

Таблица 3.3 – Переменные класса Calendar

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип данных | Назначение переменной |
| currentDate | time\_t | Хранит значение текущей даты |
| status | vector<vector<bool>> | Хранит состояние(true/false) для определенной даты |
| dates | Vector<tm> | Хранит дату на которую может быть заказана услуга |

Таблица 3.4 – Переменные класса Employee

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип данных | Назначение переменной |
| date | Calendar | Хранит график работы сотрудника |

Таблица 3.4 – Переменные класса Service

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип данных | Назначение переменной |
| name\_service | string | Хранит значение названия услуги |

Таблица 3.4 – Переменные класса Service

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип данных | Назначение переменной |
| price | float | Хранит цену за услугу |
| duration | int | Хранит длительность услуги в минутах |
| description | string | Хранит описание услуги(краткая информация) |

Таблица 3.5 – Переменные класса Order

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип данных | Назначение переменной |
| total\_price | float | Хранит цену заказа |
| services | vector<Service> | Хранит набор услуг одного заказа |
| employyes | vector<Employee> | Хранит набор данных о сотруднике |
| timeService | vector<tm> | Хранит набор о дате и времени проведения определенной услуги |
| count\_line | int | Хранит количество услуг в данном заказе |

Таблица 3.6 – Переменные класса User

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип данных | Назначение переменной |
| orders | vector<Order> | Хранит заказы пользователя |

Таблица 3.7 – Переменные класса Admin

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип данных | Назначение переменной |
| services | vector<Service> | Хранит услуги, предоставляемые салоном красоты |

Для разработки программы использовались методы. Ниже описаны классы и методы, относящиеся к соответствующему классу.

Методы класса Account:

* string getLogin();
* string getPassword();
* string getRole();
* void setRole(string role);
* void setLogin(string login);
* void setPassword(string password);
* void appUserDisplay();
* bool addLogin(vector<string> logins);
* virtual void addPassword();
* friend ifstream& operator>>(ifstream& file, Account& user).

Методы класса Calendar:

* void createCalendar();
* string getTimeString(int timeSlot) const;
* string getFormattedDate(const tm& date) const;
* tm printTime(int day, tm &time) const;
* tm printCalendar().

Методы класса Person:

* string getSurname();
* virtual void setPersonalData(int x, int y);
* virtual void getPersonalData(int x, int y).

Методы класса Employee:

* void addPassword();
* void setPersonalData(int x, int y);
* void getPersonalData(int x, int y);
* void displayEmployee();
* friend ostream& operator<<(ostream& cout, const Employee& obj).

Методы класса Order:

* void add\_Order(Service service, Employee employee, tm timeService);
* void delete\_orderInfo();
* void display\_order(int j);
* float getTotalPrice();
* friend ostream& operator<<(ostream& cout, const Order& obj).

Методы класса User:

* void setPersonalData(int x, int y);
* void getPersonalData(int x, int y);
* void add\_Service();
* void addPassword();
* void getService();
* void getEmployee();
* template<typename T>
* T addOrder(vector<T> obj);
* void addOrder(Service service, Employee employee, tm time);
* void setOrder();
* void displayOrder().

Методы класса Admin:

* void add\_Service();
* void getService();
* void addPassword().

Методы класса Service:

* Service addService();
* float getPrice();
* void displayService();
* friend ostream& operator<<(ostream& cout, const Service& obj);
* friend ifstream& operator>>(ifstream& file, Service& service).

Таким образ были рассмотрены структуры классов Account, Person, User, Admin, Employee, Service, Order, Calendar. С помощью данной структуры была разработана система автоматизации учета деятельности косметического салона. При разработке были задействованы основные принципы объектно-ориентированного программирования такие как: наследование, инкапсуляция и полиморфизм, что позволило создать гибкую и эффективную систему для автоматизации процессов учета деятельности косметического салона.

# **описание созданных программных конструкций**

Использованные конструкции были разработаны с учетом требований курсового проекта.

Реализация базовых принципов объектно-ориентированного программирования (инкапсуляции, наследования, полиморфизма) представлена на рисунках 4.1, 4.2, 4.3

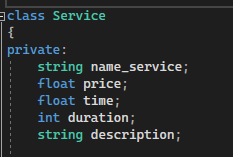


Рисунок 4.1 – Использование private полей как пример инкапсуляции



Рисунок 4.2 – Использование наследования



Рисунок 4.3 – Перегрузка функций как пример полиморфизма

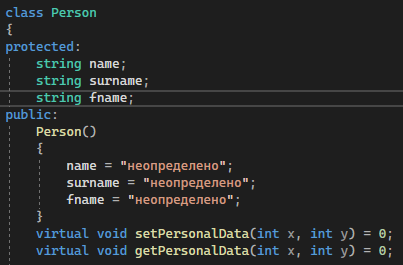
Использование абстрактных классов показано на рисунке 4.4

Рисунок 4.4 – Использование абстрактного класса

Использование передачи параметров по ссылке и по значению представлено на рисунках 4.5 и 4.6



Рисунок 4.5 – Передача параметров по ссылке



Рисунок 4.6 – Передача параметров по значению

Использование встроенных, пользовательских, дружественных и виртуальных функций представлено на рисунках 4.7, 4.8, 4.9, 4.10



Рисунок 4.7 – Использование встроенной функции

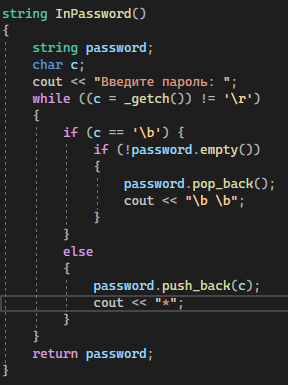


Рисунок 4.8 – Использование пользовательской функции

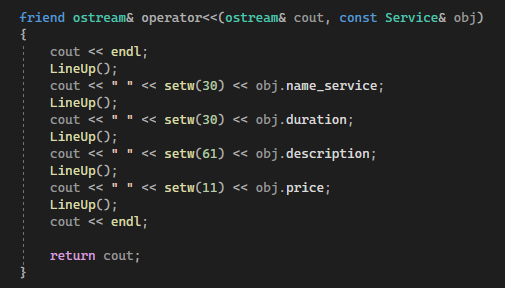


Рисунок 4.9 – Использование дружественной функции



Рисунок 4.10 – Использование виртуальных функций

Использование пространств имен встроенных и пользовательских представлено на рисунках 4.11 и 4.12.



Рисунок 4.11 – Использование встроенных пространственных имен



Рисунок 4.12 – Использование пользовательских пространственных имен

Реализация обработки ошибок программы (средствами языка си++) представлена на рисунке 4.13

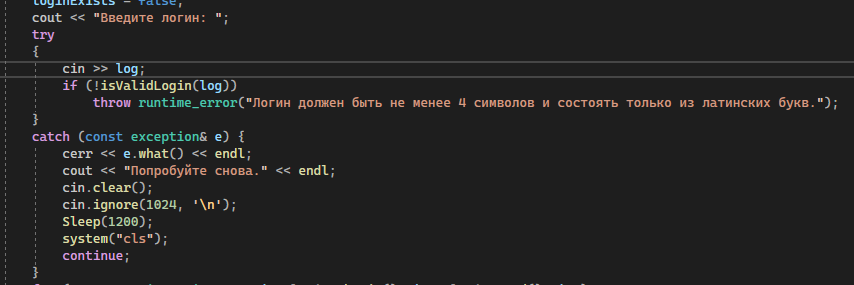


Рисунок 4.13 – Реализация обработки ошибки при вводе логина

Использование перегрузки методов и операторов, переопределение методов показана на рисунках 4.14, 4.15, 4.16

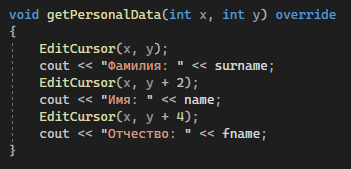


Рисунок 4.14 – Переопределение метода



Рисунок 4.15 – Перегрузка метода

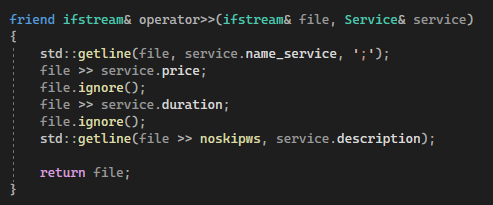


Рисунок 4.16 – Перегрузка оператора

Использование шаблонов классов и методов показано на рисунках 4.17 и 4.18



Рисунок 4.17 – Использование шаблона метода



Рисунок 4.18 – Использование шаблона класса

# **разработка и описание диаграммы классов приложения**

При разработке автоматизированной системы, ключевым этапом является проектирование её архитектуры с использованием диаграммы классов. Диаграмма классов (рисунок 5.1) предоставляет визуальное представление архитектуры системы, иллюстрируя классы и их взаимосвязи.

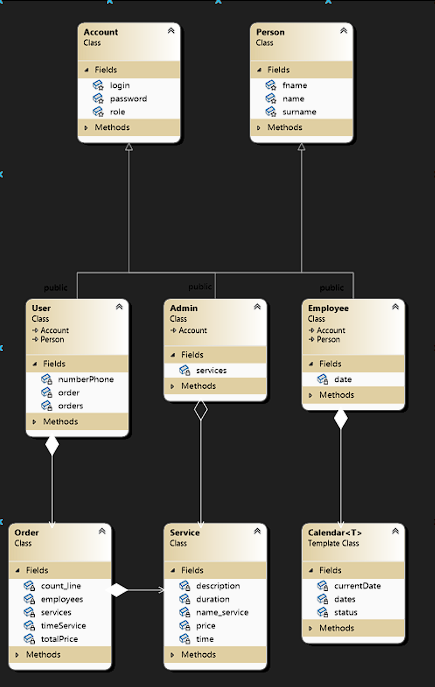


Рисунок 5.1 – Диаграмма классов

Из диаграммы классов видно, что классы User, Employee наследуются от классов Account и Person. Класс User связан с Order композицией, также как и Order связан с Service. Employee также композиция шаблонного класса Сalendar. А Admin связан с Srvice агрегацией.

# **разработка и описание диаграммы вариантов использования приложения**

Диаграмма вариантов использования — визуальное представление функциональности системы, описывающее, как различные типы пользователей взаимодействуют с системой.

В программе выделяется три основные роли (рисунок 6.1): сотрудник, администратор и пользователь, каждый из которых обладает собственным набором сценария использования приложения.

Рисунок 6.1 – Диаграмма вариантов использования

Диаграмма вариантов использования позволяет быстро определить все возможности системы относительно каждой роли, что гарантирует качественное понимания функционала системы и возможностей каждого пользователя.

# **блок-схема алгоритма всей программы и двух основных методов**

## **Блок-схема алгоритма SignIn**

# **описание алгоритма запуска приложения, Его использование, результаты работы программы, тестирование обработки ошибок**

После запуска программы пользователю предоставляется меню (рисунок 8.1), в котором он может выбрать одно из действий: Зарегистрироваться, Выйти, Войти.

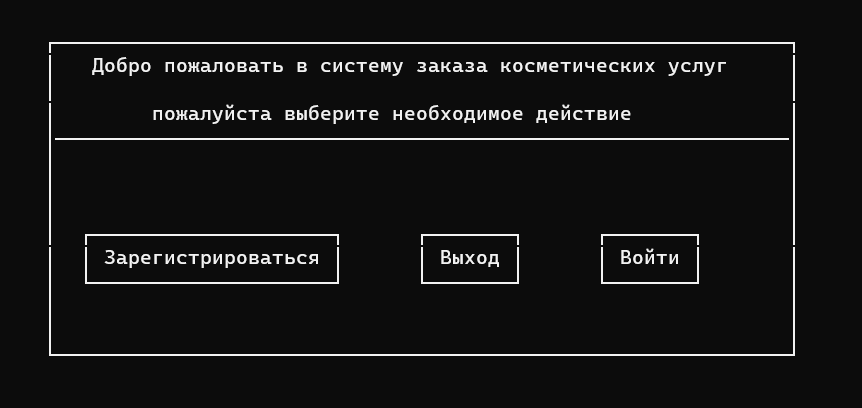


Рисунок 8.1 – Меню после запуска программы

При выборе регистрации, пользователю дается на выбор три роли (рисунок 8.2): Администратор, пользователь, сотрудник.