

## РАЗРАБОТКА СИСТЕМЫ УЧЕТА ТАЛОНОВ ВЕТЕРИНАРНОЙ КЛИНИКИ С ПРИМЕНЕНИЕМ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММИРОВАНИЯ

<sup>1</sup>Демух С. А., <sup>2</sup>Белодед Н. И.

<sup>1</sup>Академия управления при Президенте Республики Беларусь,  
Минск, Беларусь, [svetlanademikh@gmail.com](mailto:svetlanademikh@gmail.com),

<sup>2</sup>Академия управления при Президенте Республики Беларусь,  
Минск, Беларусь, [nbeloded@gmail.com](mailto:nbeloded@gmail.com)

**Аннотация.** В тезисах рассмотрена разработка консольного приложения для системы учета талонов ветеринарной клинике, показаны достоинства языка программирования C++.

Цифровая экономика активно развивается во всем мире, вовлекая в изменяющиеся процессы все большее количество стран, а также трансформируя традиционные формы производства и потребления. Происходящие перемены требуют создания принципиально новых подходов к организации рабочего пространства на предприятиях, в организациях и учреждениях.

На основе анализа существующих программных средств была разработана система учета талонов в ветеринарной клинике, которая сочетает в себе как автоматизацию, так и универсальность использования для любого работника учреждения. Основа работы – требования для систематизации непрерывно пополняющегося объема информации в случае комплексного функционирования клиники.

При рассмотрении всех требований и технологий для реализации системы выбрана концепция консольного приложения с использованием языка C++.

Название «язык программирования C++» происходит от языка программирования C, в котором унарный оператор ++ обозначает инкремент переменной. C++ сочетает свойства как высокоуровневых, так и низкоуровневых языков программирования. Главная особенность данного языка от своего предшественника – поддержка объектно-ориентированного программирования (далее – ООП), где ключевым элементом является использование классов [1].

Классы обеспечивают гарантированную инициализацию данных, сокрытие информации, динамическое определение типа, неявное преобразование определяемых пользователем типов, механизм перегрузки операторов и контроль пользователя над управлением памятью.

Классы в программировании состоят из свойств и методов. Свойства состоят из данных, которые могут охарактеризовать объект класса. Методы состоят из функций, которые могут выполнять какие-либо действия над данными (свойствами) класса, а свойства класса – его переменные.

ООП свойственны следующие принципы:

1. Наследование классов. Процесс создания классов-наследников по образу класса-родителя. Классы наследники могут наследовать все поля родительского класса и его методы.

2. Полиморфизм. Возможность объектов с одинаковой спецификацией иметь различную реализацию.

3. Инкапсуляция. Деление свойств и методов класса на доступные (public) и недоступные из вне (private, protected), при котором невозможно получить доступ к закрытым компонентам класса.

4. Абстракция. Способ выделения самых значимых характеристик объекта.

При создании объектов класса необходимо, чтобы переданные объекту значения сохранились в поля конкретного класса. Для этого используется специальная функция-конструктор – специальный метод класса, который предназначен для инициализации элементов класса некоторыми начальными значениями. Конструктор может быть реализован как по умолчанию, так и с помощью отдельной функции. Он не имеет типа возвращаемого значения и должен называться также, как класс, в котором он находится [2].

В отличие от конструктора, деструктор – специальный метод класса, который служит для уничтожения элементов класса. Чаще всего его используют тогда, когда в конструкторе при создании объекта класса динамически был выделен участок памяти и необходимо эту память очистить, если эти значения уже не нужны для дальнейшей работы программы. Имя деструктора аналогично имени конструктора, только в начале ставится знак тильды. Деструктор не имеет входных параметров.

Консольное приложение системы учета талонов в ветеринарной клинике позволит работать с двумя базами данных: пациенты и врачи. В том числе добавлять или удалять элементы базы, редактировать каталоги, записываться на прием к врачу, сортировать данные по различным параметрам, находить данные в каталоге.

В ходе работы программы используются файлы «Doctor.dat» и «Patient.dat», куда происходит запись данных о специалистах и записанных на прием пациентах. Программа включает три класса, где classDoctor является наследником classUsers, а classPatient наследником двух классов: Users и Doctor.

Для класса пациенты были созданы методы, позволяющие вводить имя, фамилию, отчество, номер телефона и животное пользователя соответственно.

Класс Doctor создан для работы с базой ветеринаров и дополнен специфическими лишь для него свойствами и методами: личным номером доктора, номером кабинета, его специализацией. С помощью методов можно добавить нового врача, его специализацию и номер кабинета.

Класс Patient позволяет записывать пациентов на прием на базе класса Doctor. Он имеет личный номер пациента, личный номер доктора для записи на прием, дата записи на прием, время записи на прием. Соответствующие методы позволяют осуществить регистрацию нового пациента, установить дату и время записи на прием.

Для работы с данными и их редактирования был разработан класс Menu, в состав которого входят следующие функции: вход и выход из базы данных,

выбор работы с врачами или пациентами, осуществление записи данных из файлов Doctor.dat, Patient.dat, осуществление записи данных в файл Doctor.dat, Patient.dat, изменение информации о врачах; поиск врача, пациента по заданным параметрам, запись подходящих клиентов и вывода списка на экран, вывод списка записанных на прием пациентов и врачей, на которых они записаны, вывод на консоль списка пациентов, проверка правильности ввода данных в поток ввода для длинных чисел, проверка правильности ввода данных в поток ввода.

Для создания программы была выбрана среда разработки Microsoft Visual Studio 2013, который является весьма удобным для создания программного продукта.

Все операции по вводу-выводу, выполнению математических операций и так далее выполняются за счет использования набора библиотечных функций, поддерживаемых компилятором. Для подключения используемых библиотек необходимо указать директиву препроцессора `#include`. В разработке данного консольного приложения были использованы следующие библиотеки:

- `#include <iomanip>` – реализация инструментов для работы с форматированием вывода;

- `#include <iostream>` – библиотека, в которой содержатся потоки ввода и вывода `cin` и `cout`;

- `#include <stdio.h>` – заголовочный файл, содержащий определения макросов, константы и объявления функций и типов для операций стандартного ввода и вывода;

- `#include <fstream>` – заголовочный файл для работы с файлами;

- `#include <Windows.h>` – подключение русского языка в консоли;

- `#include <locale.h>` – заголовочный файл, описывающий типы данных, функции и макросы, необходимы для согласования языковых аспектов;

- `#include <string>` – класс с методами и переменными для организации работы со строками.

После подключения библиотек необходимо указать необходимый элемент `using namespace std`, который открывает пространство имен библиотеки `iostream`.

В подключаемых файлах используется директива препроцессора `#pragma once`, которая применяется для защиты от «двойного подключения» заголовочных файлов.

Описание шаблонов функций и классов, используемых при работе с системой, позволяют уменьшить повторяемость кода, задав для класса универсальный тип. Перед классом указывается ключевое слово `template`, после которого идут `<>`. В `<>` после слова `typename` идет параметр шаблона.

Запись данных производится в бинарный файл и осуществляется с помощью одноименных функций.

Данное программное приложение представляет собой программу управления работой ветеринарной клиники. Вход осуществляется в режиме администратора (рис. 1).

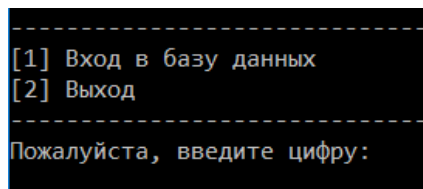


Рисунок 1 – Начало программы

После нажатия пункта «1» открывается меню администратора (рис. 2). В этом меню присутствуют функции для работы с данными врачей и пациентов, а также функция перехода в предыдущее меню.

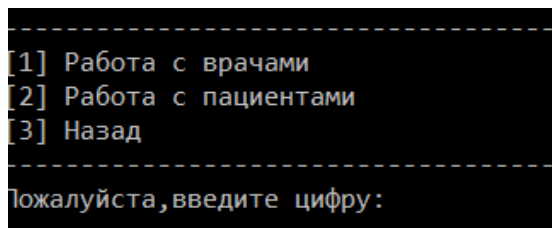


Рисунок 2 – Меню администратора

Далее нужно выбрать с каким пунктом меню будет производиться работа. Выбор пункта меню происходит нажатием клавиш «1», «2» или «3».

При выборе «1» откроется меню работы с врачами (рис. 3), при выборе «2» откроется меню работы с пациентами (рис. 4), а при выборе «3» – выход в предыдущее меню программы.

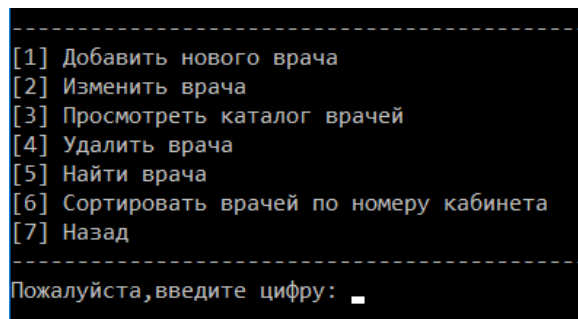


Рисунок 3 – Меню работы с врачами

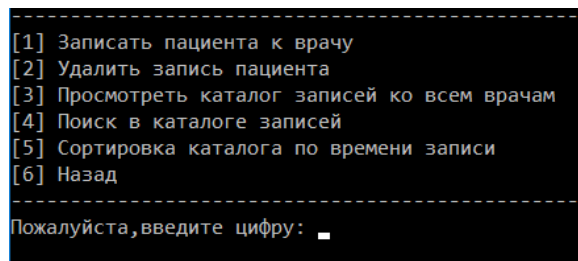


Рисунок 4 – Меню работы с пациентами

Созданная программа реализует все поставленные условия. Кроме того, данная программа работает с файлами, в том числе осуществляет чтение файлов и запись в файл. Главное меню программы предлагает ряд запросов, один из которых может выбрать пользователь. Каждый выбор обрабатывается, а после программа выдает требуемую информацию.

Благодаря гибкости и достоинствам языка С++ был получен результат работы, которым стало полноценное приложение, отвечающее за учет талонов ветеринарной клиники.

В данной разработке были использованы такие особенности языка, как классы и объекты, которые позволили создать большой и многофункциональный программный продукт.

### **Литература**

1. Белодед, Н. И. Алгоритмизация и программирование: практикум / Н. И. Белодед [и др.]. – Мн.: Академия управления при Президенте Республики Беларусь, 2005. – 163 с.

2. Джонс, Б. Освой самостоятельно С за 21 день / Б. Джонс, П. Эйткен; пер. с англ. – 6-е изд. – М.: Издательский дом «Вильямс», 2005. – 800с.: ил.

3. Интернет-сообщество Программирование для начинающих на С++ (англ. CppStudio) [Электронный ресурс]. – Режим доступа: <http://cppstudio.com>. – Дата доступа: 20.10.2022.

4. Павловская, А. Т. С/С++. Структурное программирование: практикум / А. Т. Павловская, Ю. А. Щупак. – СПб.: Питер, 2003. – 240с.

5. Уэйт, М. Язык Си: руководство для начинающих/М.Уэйт, С.Прата, Д. Мартин; пер. с англ. Л. Н. Горинович, В. С. Явнилович; под ред. Э. А. Трахтенгерца. – ThewaiteGroup, Inc, 1984.

6. Язык программирования С. Лекции и упражнения: пер. с англ. – 6-е изд. – М: ООО «И. Д. Вильямс», 2015. – 928 с.: ил.