**Содержание**

[Введение 3](#_Toc511738420)

[1 Содержание задач организации экспорта данных из ИС 4](#_Toc511738421)

[1.1 Анализ предметной области 4](#_Toc511738422)

[1.2 Анализ системы на основе диаграммы IDEF0 5](#_Toc511738423)

[2 Структура хранения документов 9](#_Toc511738424)

[2.1 Информационный фонд организации экспорта данных 9](#_Toc511738425)

[2.2 Инфологическая модель подсистемы 10](#_Toc511738426)

[2.3 Логическая модель подсистемы 10](#_Toc511738427)

[2.4 Физическая модель подсистемы 13](#_Toc511738428)

[3 Алгоритмическое и программное обеспечение обработки документов 14](#_Toc511738429)

[3.1 Формирование диаграммы UseCase 14](#_Toc511738430)

[3.2 Формирование диаграмм последовательностей 15](#_Toc511738431)

[3.3 Разработка диаграммы активности 16](#_Toc511738432)

[3.4 Описание программной реализации 17](#_Toc511738433)

[Заключение 20](#_Toc511738434)

[ПРИЛОЖЕНИЕ А](#_Toc511738435) [(справочное)](#_Toc511738436) [Листинг программы 21](#_Toc511738437)

[ПРИЛОЖЕНИЕ Б](#_Toc511738438) [(обязательное)](#_Toc511738439) [Библиографический список 34](#_Toc511738440)

Введение

В задачах проектирования и разработки сложных информационных систем одной из важных проблем является обмен данными между различными подсистемами. Зачастую простая, на первый взгляд, задача импорта/экспорта данных из одной системы в другую приводит к необходимости серьезных разработок модулей на стыке подсистем. Задача значительно облегчается, если данные определенного типа будут перемещаться между подсистемами, при условии, что в этих подсистемах будет реализована возможность воспринимать извне и отдавать наружу данные в стандартном формате импорта/экспорта. Данный подход является основой для разработки метаданных и интерфейсов для обмена регулярными данными для различных систем.

Целью данной курсовой является экспорт данных из информационной системы медкомиссии в военкомате.

1 Содержание задач организации экспорта данных из ИС

Для того, чтобы определить содержание задач экспорта данных, необходимо проанализировать предметную область. Для этого будет необходимо описать проблему и провести анализ системы на основе диаграммы IDEF0.

1.1 Анализ предметной области

Медкомиссия в военкомате — это обязательная процедура для определения годности призывника или допризывника к воинской службе. [1]

Освидетельствование граждан при постановке на воинский учет и призыве на военную службу проводят врачи-специалисты: врач-хирург, врач-терапевт, врач-невролог, врач-психиатр, врач-офтальмолог, врач-оториноларинголог, врач-стоматолог и при необходимости – врачи других специальностей. [2]

При необходимости призывника могут направить на обследование в стационар для постановки диагноза или назначить прохождение дополнительных врачей.

Призывники должны пройти осмотр у вышеперечисленных врачей, каждый из которых выставляет свою категорию годности. После прохождения призывником медицинской комиссии в военкомате выносится вердикт об единой категории его пригодности к военной службе. [1]

Таблица 1 – список категорий

|  |  |
| --- | --- |
| **Категория пригодности** | **Описание** |
| «А» | Годен к военной службе в любых войсках, включая элитные (ВДВ, пограничные, морская пехота). |
| «Б» | Годен к службе с небольшими ограничениями по выбору рода войск. Род войск обозначен цифрой (Б1, Б2, Б3, Б4). |
| «В» | Ограниченно годен к службе. Молодой человек освобождается от службы в мирное время, отправляется в запас с соответствующей отметкой в военном билете. |

Продолжение таблицы 1

|  |  |
| --- | --- |
| **Категория пригодности** | **Описание** |
| «Г» | Временно не годен. Предоставляется отсрочка по состоянию здоровья на срок 6 или 12 месяцев, предусматривается прохождение переосвидетельствования через указанный срок. |
| «Д» | Не годится. Призывник полностью комиссуется без дополнительных условий. Выдается военный билет с указанием категории годности. |

Экспорт данных позволяет использовать данные в других системах.

1.2 Анализ системы на основе диаграммы IDEF0

Методология IDEF0 предназначена для представления системы в виде набора чередующихся функций. В IDEF0 ставится акцент на соподчинённость объектов. В данной методологии рассматриваются логические отношения между функциями, а не их временная последовательность.

Описание выглядит как «чёрный ящик» с входами, выходами, управлением и механизмом, который постепенно детализируется до необходимого уровня.

Единственная функция, представленная на контекстной диаграмме верхнего уровня, может быть разложена на основные подфункции посредством создания дочерней диаграммы. В свою очередь, каждая из этих подфункций может быть разложена на составные части посредством создания дочерней диаграммы следующего, более низкого уровня, на которой некоторые или все функции также могут быть разложены на составные части. Каждая дочерняя диаграмма содержит дочерние блоки и стрелки, обеспечивающие дополнительную детализацию родительского блока.

На заглавном блоке диаграммы определена единственная функция «Учет результатов прохождения медкомиссии в военкомате» представлена на рисунке 1.

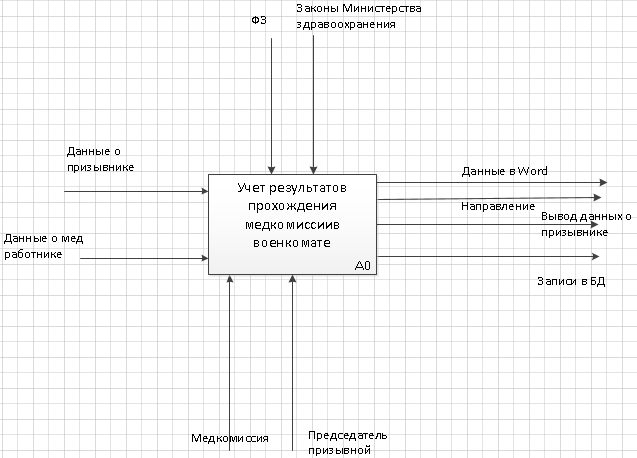


Рисунок 1 – Контекстная диаграмма системы

Основную функцию можно декомпозировать (Рисунок 2.2) на шесть блоков, которые представляют основные блоки системы:

1. «Авторизация врача»;
2. «Прохождение тестов»;
3. «Прохождение кабинетов врачей»;
4. «Прогнозирование категории»;
5. «Сохранение данных»;
6. «Экспорт данных о призывнике».

Декомпозированная контекстная диаграмма представлена на рисунке 2.

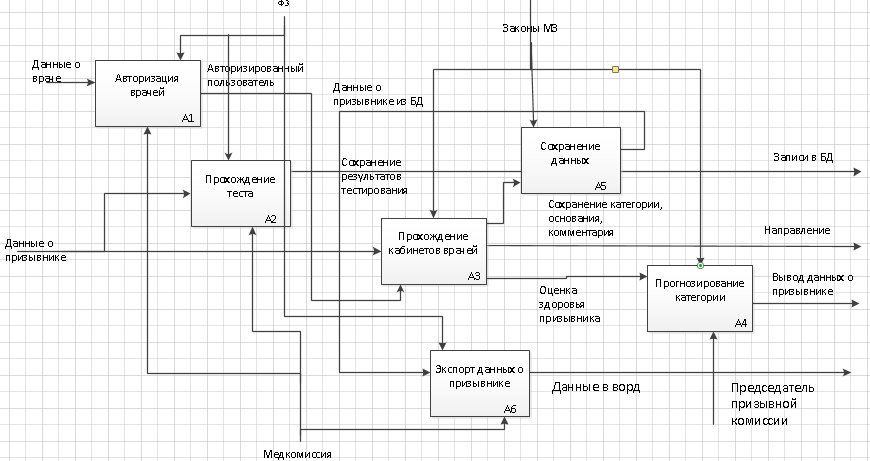


Рисунок 2 - Декомпозиция блока «Учет результатов прохождения медкомиссии в военкомате»

В данной курсовой работе нам понадобится только декомпозиция блока «Экспорт данных о призывнике».

Блок «Экспорт данных о призывнике», можно декомпозировать следующим образом, на три блока:

1. Вывод данных о призывнике.
2. Запрос в БД.
3. Вывод информации в Word.

Если на одном из этапов произошло не выполнение каких-либо требований, то система оповещает о том, что в данном случае в запросе отказано.

Декомпозиция блока «Экспорт данных о призывнике» представлена на рисунке 3.

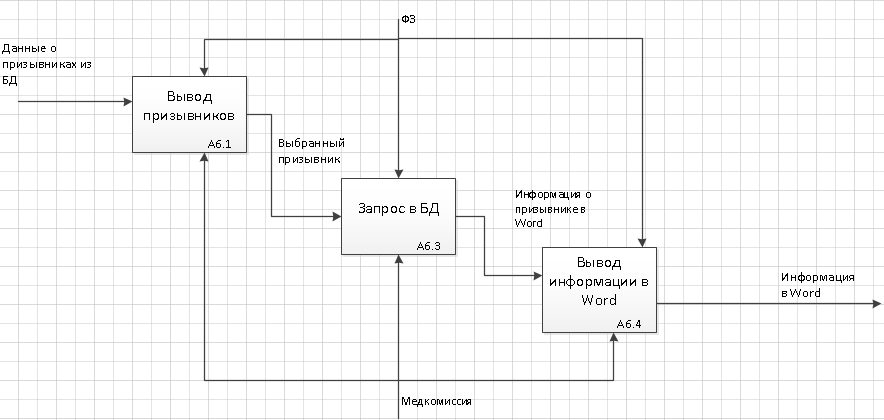


Рисунок 3 - Декомпозиция блока «Экспорт данных о призывнике»

2 Структура хранения документов

Для определения методов и средств хранения информации о врачах, призывниках необходимо проанализировать информационный фонд организации экспорта данных и спроектировать БД. Проектирование БД включает в себя разработку инфологической, логической, физической модели. Также в данной главе описаны роли пользователей и их права.

Будем называть «информационным фондом организации экспорта данных» совокупность информационных массивов, набора данных и базы данных.

2.1 Информационный фонд организации экспорта данных

Информационное обеспечение экспорта данных из информационной системы медкомиссии в военкомате предусматривает создание и функционирование единого информационного фонда, состоящего из множества информационных массивов, набора данных и базы данных.

Над элементами информационного фонда осуществляются различные операции обработки, что обеспечивает поддержание информационного фонда в актуальном состоянии, а также формируется выходная информация в соответствии с заданием на обработку.

Формирование и поддержание информационных массивов, а также все операции обработки информации осуществляются посредством подсистемы, которая организует размещение информации в базе данных и обеспечивает доступ к ней.

Для связи этих таблиц и размещения необходимой информации используется сервер баз данных MicroSoft SQL server, который также позволяет описать структуру данных как на логическом, так и физическом уровне.

Выбор базы данных как структуры хранения информации связан с тем, что системы управления базами данных предоставляют мощнейший инструментарий для создания, обновления и обработки информации, а к плюсам использования MicroSoft SQL server можно отнести высокую скорость работы и оптимальную надежность.

2.2 Инфологическая модель подсистемы

База данных представляет собой упорядоченную совокупность информации, хранящейся в виде множеств, каждое из которых содержит записи унифицированного вида.

Инфологическая модель представлена на рисунке 4.

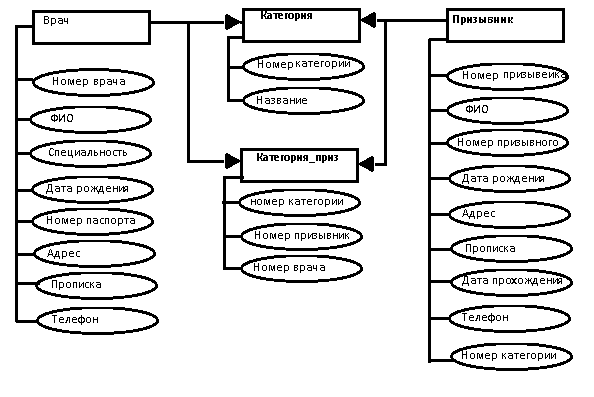


Рисунок 4 – Инфологическая модель структуры данных

2.3 Логическая модель подсистемы

Логическая модель данных описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью.

Основными компонентами логической модели являются сущности, атрибуты и связи. Каждый атрибут является характеристикой сущности,

описанием ее свойства. Связи показывают, как сущности соотносятся друг с другом логически.

Для разработки логической модели данных необходимо преобразовать инфологическую модель, представленную на рисунке 4.

Полученная логическая модель представлена на Рисунке 5.

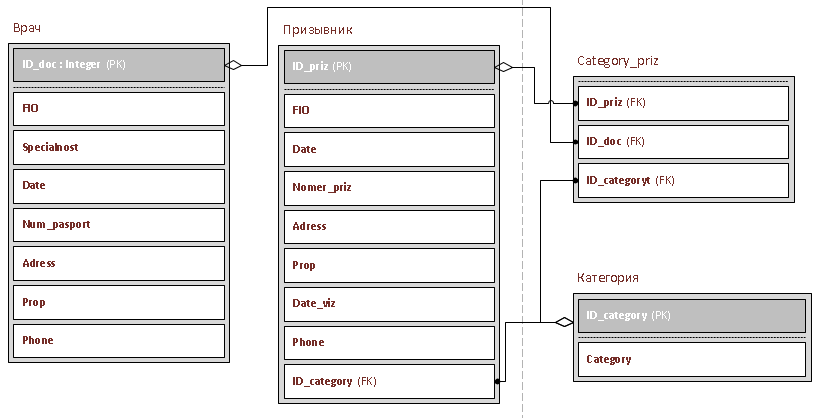


Рисунок 5 – Логическая модель данных

Описание отношений и их атрибутов представлено в таблице 2.

Таблица 2 – Характеристика атрибутов логической модели базы данных.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Отношение** | **Атрибут** | **Тип** | **Уникальность** | **Ограничения** |
| Призывник | Номер | Числовой | + | PRIMARY KEY, NOT NULL |
| ФИО | Символьный | - | NOT NULL |
| Дата рождения | Дата | - | NOT NULL |
| Адрес проживания | Символьный | - | NOT NULL |
| Прописка | Символьный | - | NOT NULL |
| Номер призывного | Числовой | + | NOT NULL |

Продолжение таблицы 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Отношение** | **Атрибут** | **Тип** | **Уникальность** | **Ограничения** |
| Призывник | Дата прохождения медкомиссии | Дата | - | NOT NULL |
| Телефон | Числовой | + | NOT NULL |
| Категория | Числовой | - | FOREIGN KEY, NOT NULL |
| Врач | Номер | Числовой | + | PRIMARY KEY, NOT NULL |
| ФИО | Символьный | - | NOT NULL |
| Дата рождения | Дата | - | NOT NULL |
| Адрес проживания | Символьный | - | NOT NULL |
| Прописка | Символьный | - | NOT NULL |
| Паспорт | Числовой | + | NOT NULL |
| Специальность | Символьный | - | NOT NULL |
| Телефон | Числовой | + | NOT NULL |
| Категория | Номер | Числовой | + | PRIMARY KEY, NOT NULL |
| Название | Символьный | + | NOT NULL |
| Кат\_приз | Номер призывника | Числовой | + | FOREIGN KEY, NOT NULL |
| Номер доктора | Числовой | + | FOREIGN KEY, NOT NULL |

Окончание таблицы 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Отношение** | **Атрибут** | **Тип** | **Уникальность** | **Ограничения** |
| Кат\_приз | Номер категории | Числовой | + | FOREIGN KEY, NOT NULL |

2.4 Физическая модель подсистемы

Физический уровень данных – это отображение системного каталога, который зависит от конкретной реализации системы управления базой данных.

На физическом уровне объекты базы данных могут называться так, как того требуют ограничения системы управления базой данных.

Физическая модель данных представлена на Рисунке 6.

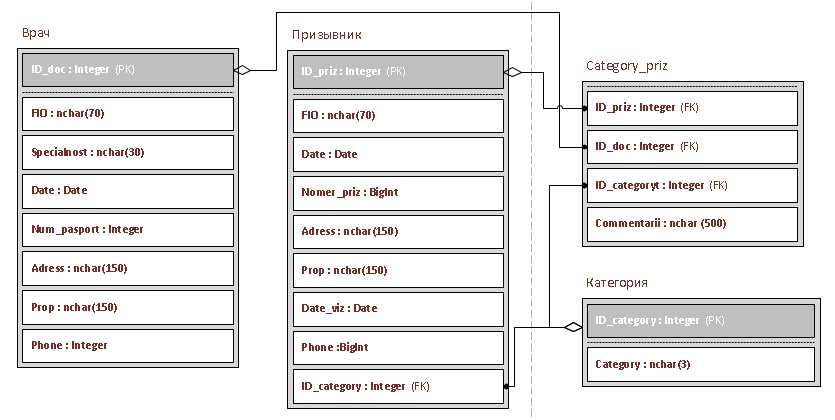


Рисунок 6 – Физическая модель данных

Скрипт для реализации базы данных приведен в приложении А.

3 Алгоритмическое и программное обеспечение обработки документов

Объектно-ориентированное проектирование (ООП) — подход к проектированию ИС, при котором система разбивается на набор объектов, соответствующих объектам реального мира и взаимодействующих между собой путем посылки сообщений. Модели UML помогают понять, обсудить и спроектировать системы программного обеспечения. MS Visio предоставляет шаблоны для наиболее часто используемых схем UML:

* Use case diagram (диаграммы вариантов использования);
* Activity diagram (диаграммы активности);
* Sequence diagram (диаграммы последовательностей действий);
* Class diagram (диаграммы классов).

3.1 Формирование диаграммы UseCase

Диаграмма вариантов использования описывает функциональное назначение системы, то есть то, что система будет делать в процессе своего функционирования. Вариант использования описывает взаимодействие системы с её окружением, инициируемое извне. Вариант использования описывает типичное взаимодействие между пользователем и ПО. Каждый вариант использования связан с целью какого-либо пользователя, которая может быть достигнута в ходе его выполнения. На диаграммах варианты использования представляются в виде овалов [3].

Действующее лицо – это роль, которую пользователь или внешняя программная система играет по отношению к системе. На диаграммах вариантов использования они изображаются в виде стилизованных человеческих фигурок. Действующим лицом может быть пользователь-человек, внешняя программная система или время, если запуск каких-либо событий в системе зависит от времени [3].

Действующим лицом диаграммы вариантов использования в разрабатываемой системе является посетитель, поскольку именно он работает с системой. Необходимо предусмотреть следующие варианты использования:

* ознакомиться с афишей музея;
* получить информацию об экспонате;
* пройти игру/викторину;
* получить информацию о музее и его истории.

Диаграмма вариантов использования представлена на рисунке 7.

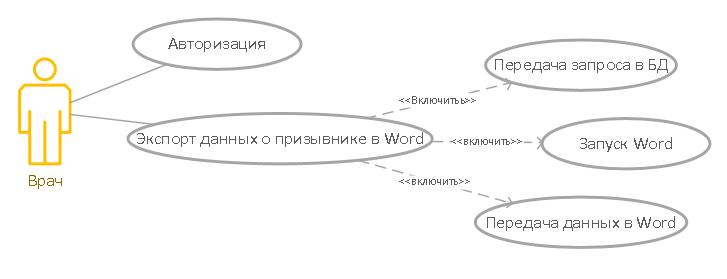


Рисунок 7 – Диаграмма Use Case

3.2 Формирование диаграмм последовательностей

Диаграмма последовательности действий отображает взаимодействие объектов, упорядоченное по времени [3].

Основными компонентами данной диаграммы являются:

* объекты;
* линии жизни;
* сообщения.

Объект является экземпляром класса.

Цель построения данной диаграммы – описать жизненный цикл объектов системы.

Диаграмма последовательности строится для каждого варианта использования [3].

Диаграмма последовательности для варианта использования «Экспорт данных о призывнике в Word» представлена на рисунке 8.

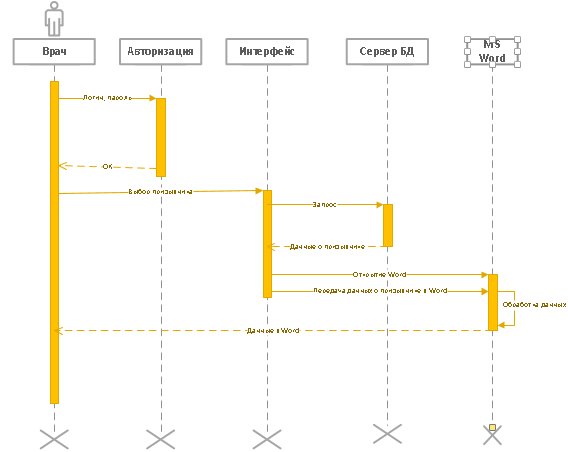


Рисунок 8 – Диаграмма последовательностей «Экспорт данных о призывнике в Word»

3.3 Разработка диаграммы активности

Диаграмма деятельности отражает динамику системы и представляет собой схемы потоков управления в системе от действия к действию, а также параллельные действия и альтернативные потоки [3].

Основные элементы диаграмм деятельности:

* деятельность (действие);
* переход;
* элемент выбора;
* линия синхронизации (линейка синхронизации).

Диаграмма активности для варианта использования «Экспорт данных о призывнике» представлена на рисунке 9.

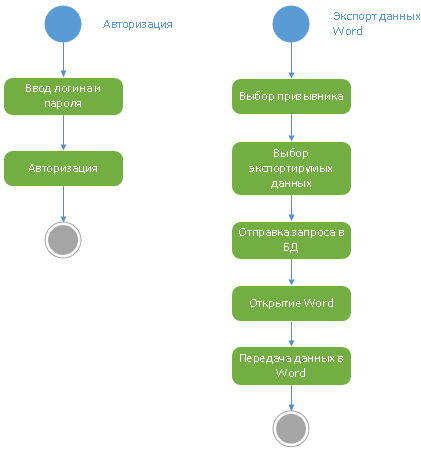


Рисунок 9 – Диаграмма активности

3.4 Описание программной реализации

В качестве языка программирования для разработки был выбран язык программирования C#, являющимся одним из самых востребованных языков.

При запуске приложения мы попадаем на основную форму рисунок 10.

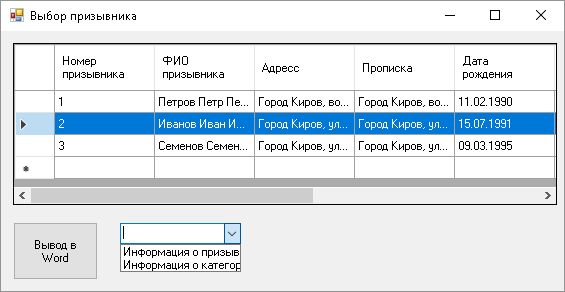


Рисунок 10 – Основная форма

Результаты приведены на рисунке 11

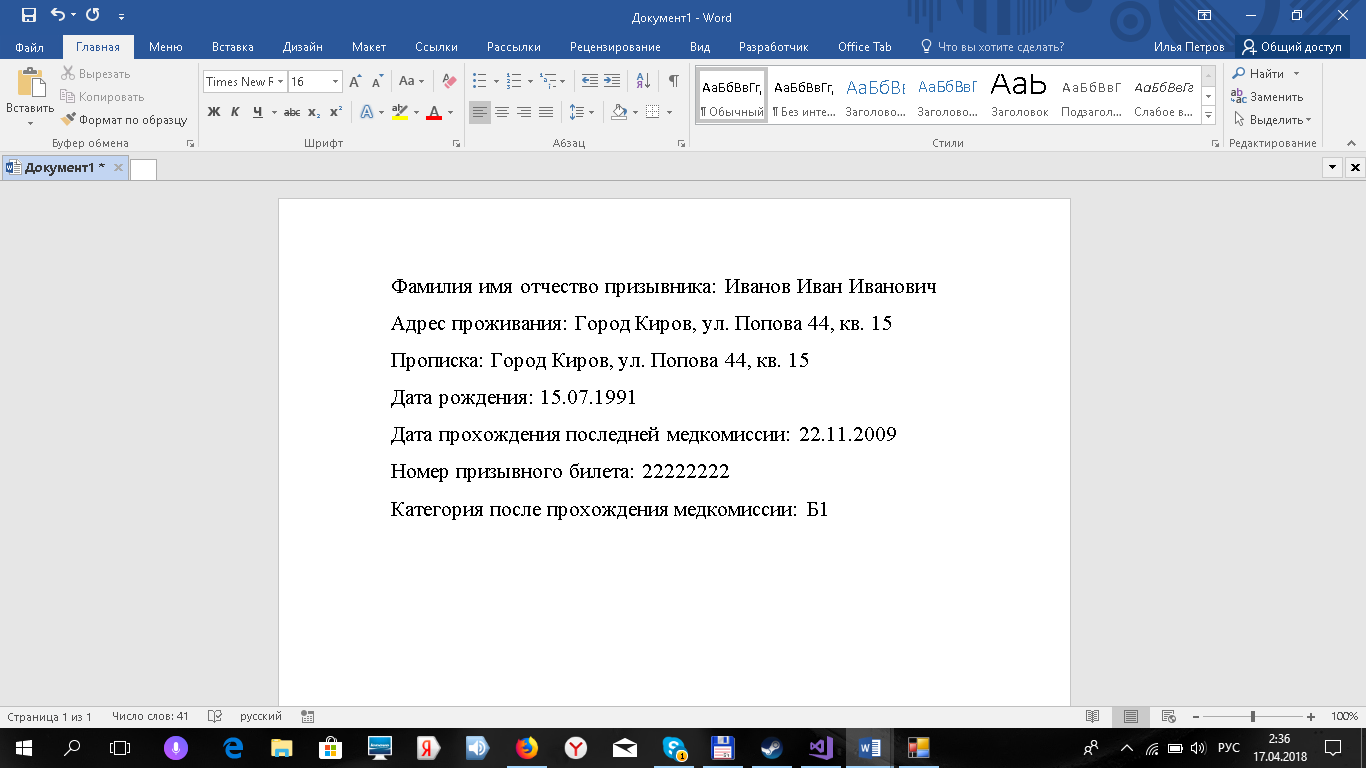


Рисунок 11 – Результат выполнения «Вывод данных о призывниках»

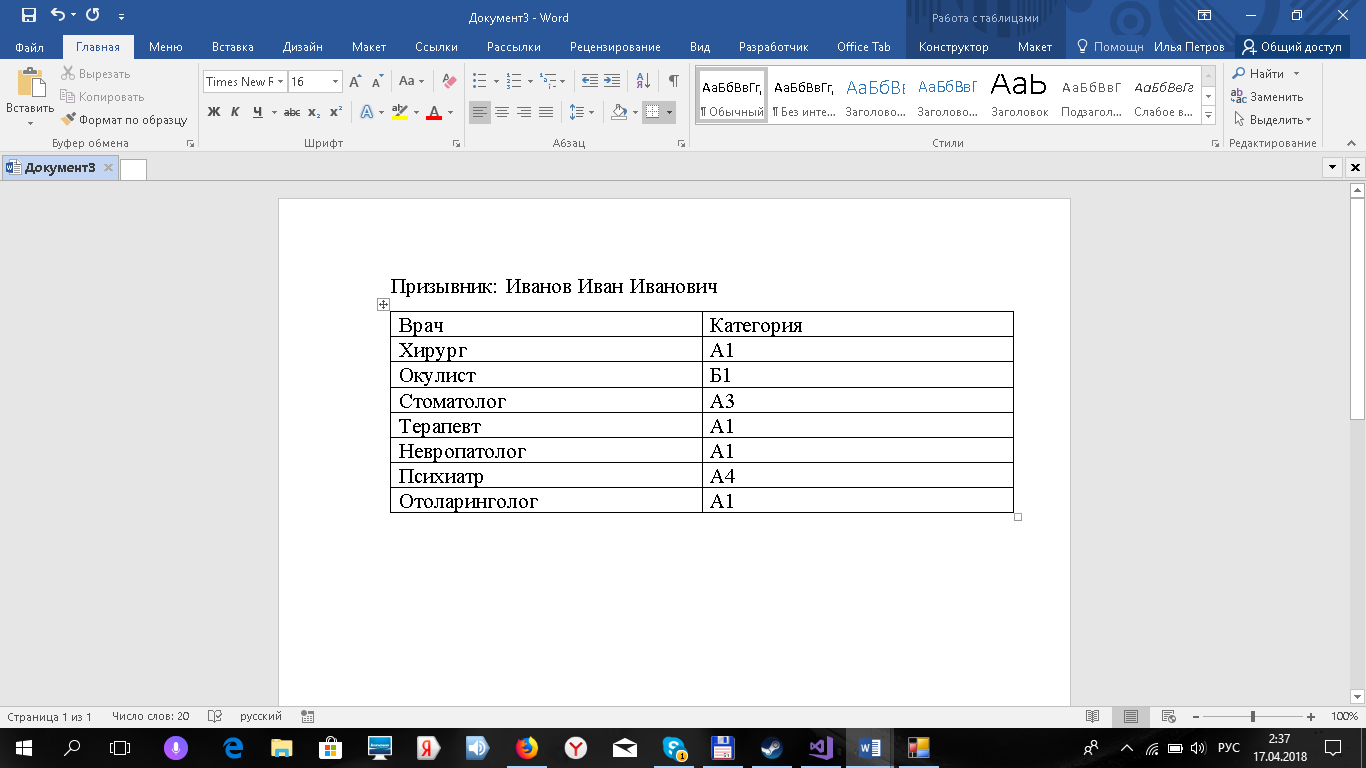


Рисунок 12 – Результат выполнения «Вывод данных о категориях»

Код программной реализации приведен в Приложении А.

Заключение

В рамках выполнения курсовой работы были выполнены следующие задачи:

* Описаны функции системы
* Спроектирована БД для хранения информации о врачах, призывниках и категориях.
* Спроектированы UML-диаграммы.
* Выполнена программная реализация на языке C#

ПРИЛОЖЕНИЕ А   
(справочное)   
Листинг программы

Скрипт создания БД:

GO

/\*\*\*\*\*\* Object: Table [dbo].[Prizivnik] Script Date: 16.04.2018 22:36:11 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Prizivnik](

[ID\_prizivnik] [int] NOT NULL,

[FIO] [nchar](70) NOT NULL,

[Date] [date] NOT NULL,

[Nomer\_priz] [int] NOT NULL,

[Adress] [nchar](150) NOT NULL,

[Prop] [nchar](150) NOT NULL,

[Date\_viz] [date] NULL,

[Phone] [int] NOT NULL,

[ID\_category] [int] NOT NULL,

CONSTRAINT [PK\_Prizivnik] PRIMARY KEY CLUSTERED

(

[ID\_prizivnik] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Rout] Script Date: 16.04.2018 22:36:11 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Rout](

[ID\_rout] [int] NOT NULL,

[Name] [nchar](50) NOT NULL,

[File\_name] [nchar](25) NOT NULL,

CONSTRAINT [PK\_Rout] PRIMARY KEY CLUSTERED

(

[ID\_rout] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Specialnost] Script Date: 16.04.2018 22:36:11 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Specialnost](

[ID\_specialnost] [int] NOT NULL,

[specialnost] [nchar](10) NULL,

CONSTRAINT [PK\_Specialnost] PRIMARY KEY CLUSTERED

(

[ID\_specialnost] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

/\*\*\*\*\*\* Object: Table [dbo].[Test] Script Date: 16.04.2018 22:36:11 \*\*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[Test](

[ID\_test] [int] NOT NULL,

[Question] [nchar](300) NOT NULL,

CONSTRAINT [PK\_Test] PRIMARY KEY CLUSTERED

(

[ID\_test] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[Analys\_priz] WITH CHECK ADD CONSTRAINT [FK\_Analys\_priz\_Analys] FOREIGN KEY([ID\_analys])

REFERENCES [dbo].[Analys] ([ID\_analys])

GO

ALTER TABLE [dbo].[Analys\_priz] CHECK CONSTRAINT [FK\_Analys\_priz\_Analys]

GO

ALTER TABLE [dbo].[Analys\_priz] WITH CHECK ADD CONSTRAINT [FK\_Analys\_priz\_Prizivnik] FOREIGN KEY([ID\_priz])

REFERENCES [dbo].[Prizivnik] ([ID\_prizivnik])

GO

ALTER TABLE [dbo].[Analys\_priz] CHECK CONSTRAINT [FK\_Analys\_priz\_Prizivnik]

GO

ALTER TABLE [dbo].[Category\_priz] WITH CHECK ADD CONSTRAINT [FK\_Category\_priz\_Category] FOREIGN KEY([ID\_category])

REFERENCES [dbo].[Category] ([ID\_category])

GO

ALTER TABLE [dbo].[Category\_priz] CHECK CONSTRAINT [FK\_Category\_priz\_Category]

GO

ALTER TABLE [dbo].[Category\_priz] WITH CHECK ADD CONSTRAINT [FK\_Category\_priz\_Doc] FOREIGN KEY([ID\_doc])

REFERENCES [dbo].[Doc] ([ID\_doc])

GO

ALTER TABLE [dbo].[Category\_priz] CHECK CONSTRAINT [FK\_Category\_priz\_Doc]

GO

ALTER TABLE [dbo].[Category\_priz] WITH CHECK ADD CONSTRAINT [FK\_Category\_priz\_Prizivnik] FOREIGN KEY([ID\_priz])

REFERENCES [dbo].[Prizivnik] ([ID\_prizivnik])

GO

ALTER TABLE [dbo].[Category\_priz] CHECK CONSTRAINT [FK\_Category\_priz\_Prizivnik]

GO

ALTER TABLE [dbo].[Commentarii] WITH CHECK ADD CONSTRAINT [FK\_Commentarii\_Doc] FOREIGN KEY([ID\_doc])

REFERENCES [dbo].[Doc] ([ID\_doc])

GO

ALTER TABLE [dbo].[Commentarii] CHECK CONSTRAINT [FK\_Commentarii\_Doc]

GO

ALTER TABLE [dbo].[Commentarii] WITH CHECK ADD CONSTRAINT [FK\_Commentarii\_Prizivnik] FOREIGN KEY([ID\_priz])

REFERENCES [dbo].[Prizivnik] ([ID\_prizivnik])

GO

ALTER TABLE [dbo].[Commentarii] CHECK CONSTRAINT [FK\_Commentarii\_Prizivnik]

GO

ALTER TABLE [dbo].[Diag\_priz] WITH CHECK ADD CONSTRAINT [FK\_Diag\_priz\_Diagnoz] FOREIGN KEY([ID\_diagnoz])

REFERENCES [dbo].[Diagnoz] ([ID\_diag])

GO

ALTER TABLE [dbo].[Diag\_priz] CHECK CONSTRAINT [FK\_Diag\_priz\_Diagnoz]

GO

ALTER TABLE [dbo].[Diag\_priz] WITH CHECK ADD CONSTRAINT [FK\_Diag\_priz\_Doc] FOREIGN KEY([ID\_doc])

REFERENCES [dbo].[Doc] ([ID\_doc])

GO

ALTER TABLE [dbo].[Diag\_priz] CHECK CONSTRAINT [FK\_Diag\_priz\_Doc]

GO

ALTER TABLE [dbo].[Diag\_priz] WITH CHECK ADD CONSTRAINT [FK\_Diag\_priz\_Prizivnik] FOREIGN KEY([ID\_priz])

REFERENCES [dbo].[Prizivnik] ([ID\_prizivnik])

GO

ALTER TABLE [dbo].[Diag\_priz] CHECK CONSTRAINT [FK\_Diag\_priz\_Prizivnik]

GO

ALTER TABLE [dbo].[Doc] WITH CHECK ADD CONSTRAINT [FK\_Doc\_Specialnost] FOREIGN KEY([ID\_specialnost])

REFERENCES [dbo].[Specialnost] ([ID\_specialnost])

GO

ALTER TABLE [dbo].[Doc] CHECK CONSTRAINT [FK\_Doc\_Specialnost]

GO

ALTER TABLE [dbo].[Otvet] WITH CHECK ADD CONSTRAINT [FK\_Otvet\_Prizivnik] FOREIGN KEY([ID\_priz])

REFERENCES [dbo].[Prizivnik] ([ID\_prizivnik])

GO

ALTER TABLE [dbo].[Otvet] CHECK CONSTRAINT [FK\_Otvet\_Prizivnik]

GO

ALTER TABLE [dbo].[Otvet] WITH CHECK ADD CONSTRAINT [FK\_Otvet\_Test] FOREIGN KEY([ID\_test])

REFERENCES [dbo].[Test] ([ID\_test])

GO

ALTER TABLE [dbo].[Otvet] CHECK CONSTRAINT [FK\_Otvet\_Test]

GO

ALTER TABLE [dbo].[Prizivnik] WITH CHECK ADD CONSTRAINT [FK\_Prizivnik\_Category] FOREIGN KEY([ID\_category])

REFERENCES [dbo].[Category] ([ID\_category])

GO

ALTER TABLE [dbo].[Prizivnik] CHECK CONSTRAINT [FK\_Prizivnik\_Category]

Листинг программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

using System.Windows.Forms;

using Word = Microsoft.Office.Interop.Word;

namespace WindowsFormsApp3

{

public partial class Form1 : Form

{

public static string k;

public static int p;

public Form1()

{

InitializeComponent();

open();

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

k = dataGridView1.CurrentCell.Value.ToString();

p = dataGridView1.CurrentRow.Index;

}

public void open()

{

try

{

this.prizivnikTableAdapter.FillBy(this.vKRDataSet.Prizivnik);

dataGridView1.DataSource = prizivnikTableAdapter.GetData();

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

}

private void button1\_Click(object sender, EventArgs e)

{

Word.Application appWord = new Word.Application();

switch (comboBox1.SelectedIndex)

{

case 0:

{

string fio, adr, prop, data, dataviz, nomer;

DataRow [] cat;

int categoryid;

DataTable tablecat = new VKRDataSet.CategoryDataTable();

fio = dataGridView1.Rows[p].Cells[1].Value.ToString();

adr = dataGridView1.Rows[p].Cells[2].Value.ToString();

prop = dataGridView1.Rows[p].Cells[3].Value.ToString();

data = dataGridView1.Rows[p].Cells[4].Value.ToString();

dataviz = dataGridView1.Rows[p].Cells[6].Value.ToString();

nomer = dataGridView1.Rows[p].Cells[5].Value.ToString();

try

{

this.categoryTableAdapter1.Category(this.vKRDataSet.Category);

tablecat = categoryTableAdapter1.GetData();

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

categoryid = System.Convert.ToInt32(dataGridView1.Rows[p].Cells[8].Value.ToString());

cat = tablecat.Select("ID\_category="+categoryid);

appWord.Visible = true;

var doc = appWord.Documents.Add();

doc.Application.Run("InformationPriz", fio, adr, prop, data, dataviz, nomer, cat[0][1].ToString());

break;

}

case 1:

{

DataTable tablecat = new VKRDataSet.CategoryDataTable();

DataTable tableprizcat = new VKRDataSet.Category\_prizDataTable();

string fio;

string[] category = new string[7];

int prizid;

int g;

DataRow[] cat;

DataRow[] catpriz;

try

{

this.categoryTableAdapter1.Category(this.vKRDataSet.Category);

tablecat = categoryTableAdapter1.GetData();

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

prizid = System.Convert.ToInt32(dataGridView1.Rows[p].Cells[0].Value.ToString());

fio = dataGridView1.Rows[p].Cells[1].Value.ToString();

try

{

this.category\_prizTableAdapter1.catpriz(this.vKRDataSet.Category\_priz);

tableprizcat = category\_prizTableAdapter1.GetData();

}

catch (System.Exception ex)

{

System.Windows.Forms.MessageBox.Show(ex.Message);

}

catpriz = tableprizcat.Select("ID\_priz=" + prizid);

for (int i = 0; i < catpriz.GetLength(0); i++)

{

g = System.Convert.ToInt32(catpriz[i][1]);

cat = tablecat.Select("ID\_category=" + g);

category[i] = cat[0][1].ToString();

}

appWord.Visible = true;

var doc = appWord.Documents.Add();

doc.Application.Run("InformationCat", fio, category[0],category[1], category[2], category[3], category[4], category[5], category[6]);

break;

}

default:

{

break;

}

}

Листинг макросов:

Sub InformationPriz(ByVal fio As String, ByVal adres As String, ByVal prop As String, ByVal data As String, ByVal dataviz As String, ByVal nomer As String, ByVal cat As String)

'

' InformationPriz Макрос

'

'

Selection.Font.Size = 16

Selection.Font.Name = "Times New Roman"

Selection.TypeText Text:="Фамилия имя отчество призывника: " + fio

Selection.TypeParagraph

Selection.TypeText Text:="Адрес проживания: " + adres

Selection.TypeParagraph

Selection.TypeText Text:="Прописка: " + prop

Selection.TypeParagraph

Selection.TypeText Text:="Дата рождения: " + data

Selection.TypeParagraph

Selection.TypeText Text:="Дата прохождения последней медкомиссии: " + dataviz

Selection.TypeParagraph

Selection.TypeText Text:="Номер призывного билета: " + nomer

Selection.TypeParagraph

Selection.TypeText Text:="Категория после прохождения медкомиссии: " + cat

End Sub

Sub InformationCat(ByVal fio As String, ByVal oku As String, ByVal ter As String, ByVal sto As String, ByVal hir As String, ByVal nev As String, ByVal psi As String, ByVal oto As String)

'

' InformationCat Макрос

'

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:="Призывник: " + fio

Selection.TypeParagraph

ActiveDocument.Tables.Add Range:=Selection.Range, NumRows:=8, NumColumns:= \_

2, DefaultTableBehavior:=wdWord9TableBehavior, AutoFitBehavior:= \_

wdAutoFitFixed

With Selection.Tables(1)

If .Style <> "Сетка таблицы" Then

.Style = "Сетка таблицы"

End If

.ApplyStyleHeadingRows = True

.ApplyStyleLastRow = False

.ApplyStyleFirstColumn = True

.ApplyStyleLastColumn = False

.ApplyStyleRowBands = True

.ApplyStyleColumnBands = False

End With

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:="Врач"

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:="Категория"

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:="Окулист"

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:=oku

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:="Терапевт"

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:=ter

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:="Стоматолог"

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:=sto

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:="Хирург"

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:=hir

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:="Невропатолог"

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:=nev

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:="Психиатр"

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:=psi

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:="Отоларинголог"

Selection.MoveRight Unit:=wdCell

Selection.Font.Name = "Times New Roman"

Selection.Font.Size = 16

Selection.TypeText Text:=oto

End Sub

ПРИЛОЖЕНИЕ Б   
(обязательное)   
Библиографический список

1. Служба помощи призывникам [Электронный ресурс] URL: https://armyhelp.ru/prizyiv-v-armiyu-v-2018-godu/medkomissiya-v-voenkomate (дата обращения 23.02.2018).
2. КонсультантПлюс [Электронный ресурс] URL: http://www.consultant.ru/document/cons\_doc\_LAW\_149096/b29a8a7e90d241e0f4eb4bbff0d993e7b49cfa9b/ (дата обращения 23.02.2018).
3. Технологии разработки программного обеспечения: Учебник для вузов. 3-е изд./ С.А. Орлов. – СПб.: Питер, 2004. – 527 с.