

**УНИВЕРСИТЕТ ПО БИБЛИОТЕКОЗНАНИЕ И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ**

**КАТЕДРА “ИНФОРМАЦИОННИ СИСТЕМИ И ТЕХНОЛОГИИ”**

**СПЕЦИАЛНОСТ “ИНФОРМАТИКА И КОМПЮТЪРНИ НАУКИ”**

УЧЕБНА ДИСЦИПЛИНА: СТРУКТУРИ ОТ ДАННИ И АЛГОРИТМИ

**КУРСОВА РАБОТА**

на тема:

**Система за контрол на работното време**

Изготвил: Преподавател:

Борис Божилов проф. Иван Иванов

“Информатика и компютърни науки"

II курс, II семестър, редовно обучение

Ф № 46261р

София

2020 г.

**Съдържание**

[**Въведение** 3](#_Toc41945687)

[**1. Описание на системата** 4](#_Toc41945688)

[**1.1.** **Приложението Work Time Control System** 5](#_Toc41945689)

[**1.1.1. Формата Firms** 8](#_Toc41945690)

[**1.1.2.** **Формата Employees** 10](#_Toc41945691)

[**1.1.3.** **Формата Attendance** 11](#_Toc41945692)

[**1.1.4.** **Формата Monthly report** 12](#_Toc41945693)

[**1.2.** **Приложението Work Attendance Evidence** 14](#_Toc41945694)

[**2.** **Описание на базата данни** 16](#_Toc41945695)

[**3.** **Инсталация на системата** 17](#_Toc41945696)

[**Заключение** 18](#_Toc41945697)

# **Въведение**

Контролът и прецизното отчитане на работното време са съществен проблем за голяма част от организациите. Използването на система за контрол на работното време е интелигентно и икономично решение на тези проблеми, за всяка организация и бизнес.

Целта на разработенето приложение, не е да реши тази комплексна проблематика, която по сложеност многократно надвишава обема на една курсова работа, а да покаже основните компоненти на една такава система и да послужи за демонстрация на знанията и уменията за разработка на приложения в съвременните визуални среди за програмиране, както и **умения за употреба на структури от данни заложени в езика C#**.

Структурите от данни и алгоритмите стоят в основата на програмирането. Това са най-важните фундаментални знания на един програмист. След като се овладеят добре, няма никакви проблеми да се овладее която и да е софтуерна технология, библиотека, framework или програмен интерфейс. Именно на това разчитат и най-сериозните софтуерни фирми в света, когато наемат програмисти. Потвърждение са интервютата в големите фирми като Google и Microsoft, които изключително много държат на правилното **алгоритмично мислене** и **познаването на всички базови структури от данни и алгоритми**.

Изборът на структура данни, която ще се използва за дадена функционалност започва с изброяване на ключовите операции, които ще се извършват върху нея. След това се анализират възможните структури, които могат да бъдат използвани, и от тях се избира тази, **която най-лесно и ефективно реализира тези операции**. Често се прави **компромис между леснотата на реализация и ефективността**.

# **Описание на системата**

Дадената система по своята архитектура, макър и доста проста, представлява клиент-сървър система с две приложения. Замислено е базата данни да се намира на MS SQL Server, който е достъпен в локалната мрежа.

Системата се състои от две приложения:

1. **Work Time Control System** – представява администраторската част на системата. Тук се въвеждат данните за фирмите поместени в сградата, данните за служителите на фирмите и се извличат справки за работното време на служителите на дадена фирма за дадена дата и се изчислява работното време за целия месец.
2. **Work Attendance Evidence** – приложение за отчитане на пристигането и тръгването от работа, което е предназначено да работи на отделен компютър, намиращ се на входната врата на сградата.

Двете приложения независимо достъпват базата данни, която може да бъде качена и на отдалечен сървър в Интернет.

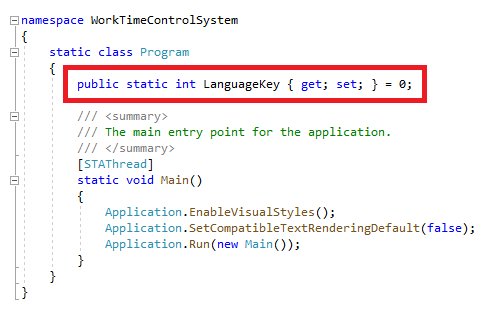
Приложенията са с **мултиезичен интерфейс**, **реализиран с използването на вградената структура от данни – речник** (в комбинация със вложен списък), с възможност за избор на един от три езици: **английски, български и сръбски**.

## **Приложението Work Time Control System**

Приложението Work Time Control System се състои от пет форми. При стартиране на приложението първоначално се зарежда главната форма на приложението, от която се осъществява връзка към останалите четири форми. В главната форма има четири командни бутона, чрез които се извикват другите форми, а във върха на формата са расположени и три командни бутона (с флаговете на съответните държави), чрез които се осъществява превключването на езика на приложението.



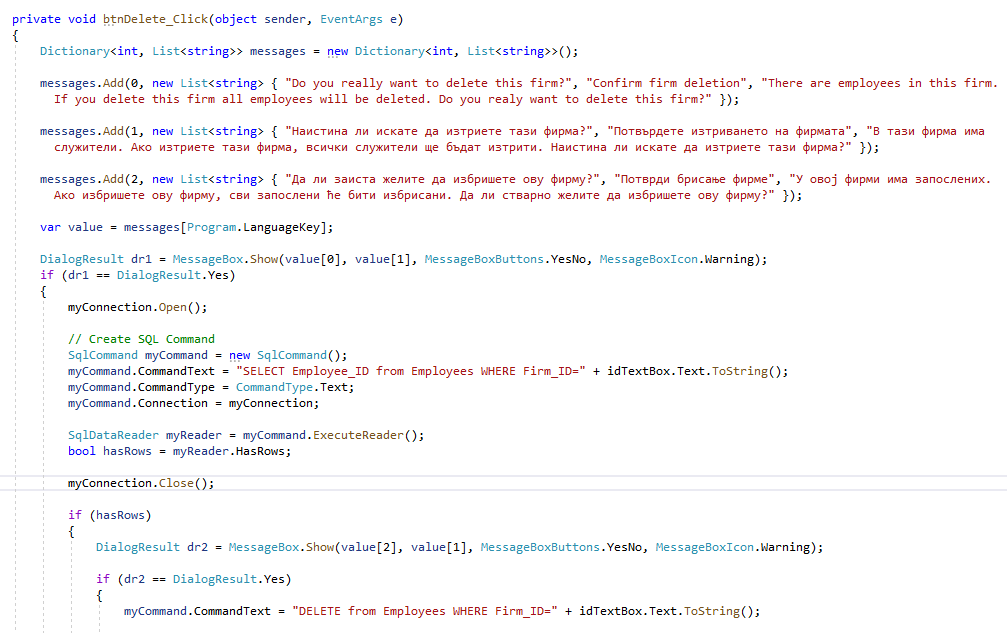
Фигура 1 – Начална форма на приложението на английски език

В продължение, **за илюстрация** е даден примерен код за смяна на езика на приложението.

В приложението е деклалирана глобална променлива **LanguageKey**, с помощта на която се превключва езика на интерфейса.

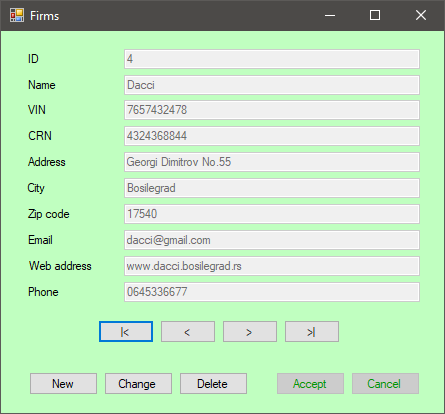
За смяната на езика на контролите е използван речник с вложени в него като стойности списъци, съдържащи названията на контролите на съответния език, а ключове са имената на контролите във формата.

С помощта на променливата **LanguageKey** се извлича названието на контролата на съотвентия език.

Превключването на езика в Message Box-овете се реализира чрез списък, чийто ключ е променливата **LanguageKey** и стойности, които представляват съобщенията на съответния език.

### **Формата Firms**

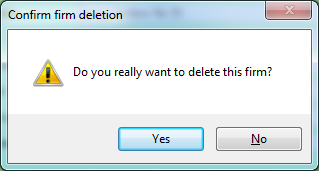
В тази форма се въвеждат данните за фирмите, които са поместени в сградата. Тя се състои от десет TextBox - а, бутон за въвеждане на нова фирма, бутон за ъпдейтване на данните за съществуваща фирма, бутон за изтриване на съществуваща фирма, както и от бутони за навигация, с помощта на които се придвижва по записите в базата.



Фигура 2 – Формата на Firms

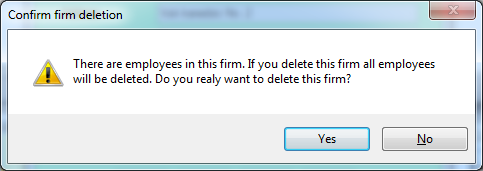
Инициално полетата и бутоните Accept и Cancel са неактивни. При натискане на бутоните New или Change, полетата се активират и овъзможават въвеждане или изменение на данните, бутонът Cancel също се активира, а бутоните New, Change и Delete, както и бутоните за навигация стават неактивни. Всички полета имат валидация на данните, което предпазва базата от грешно въведени данни от страна на потребителя. Само при правилно въведени данни бутона Accept става активен и при натискане на бутона данните се записват на сървъра. При натискане на бутона Cancel действията се отменят и отново се зареждат данните от сървъра .

При опит да се изтрие дадена фирма се появява MessageBox с контролен въпрос:



Фигура 3 – Диалогов прозорец с контролен въптрос за изтриване на фирма

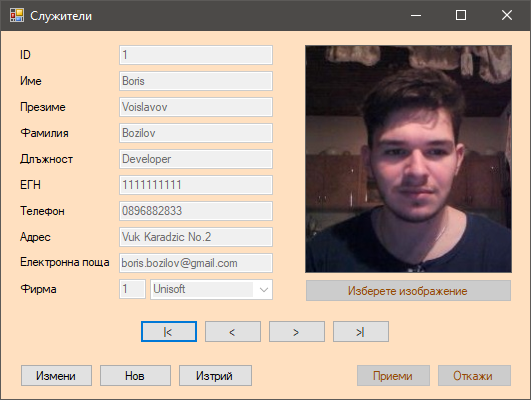
При потвърждение се прави проверка дали дадената фирма има въведени заети лица. В случай че няма заети, фирмата се изтрива. В противен случай се активира нов MessageBox с предупреждение, че покрай фирмата ще бъдат изтрити и всички заети. Това е необходимо за запазване на референциалния интегритет на базата. Именно, първо се изтриват заетите, а след това и фирмата.



Фигура 4 – MessageBox с предупреждение и повторно потвърждение

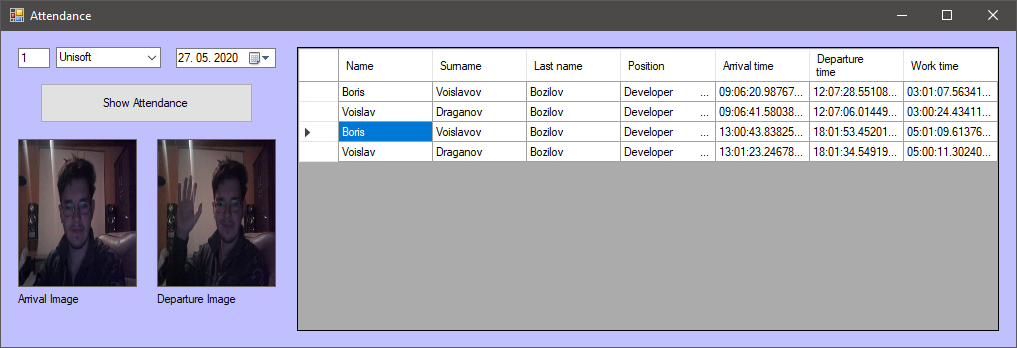
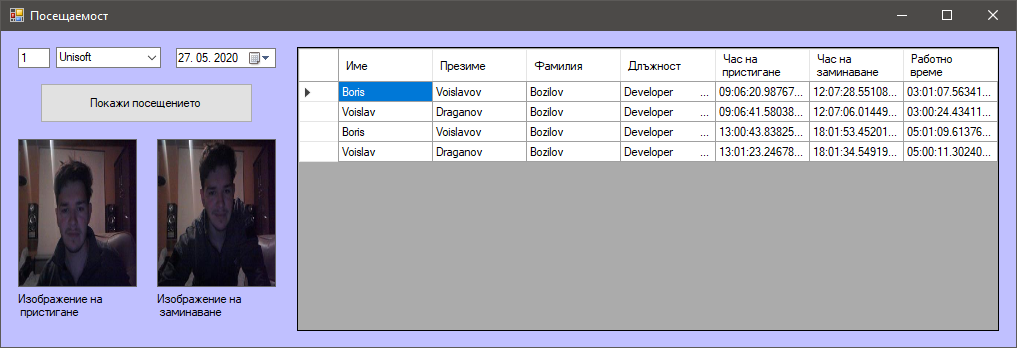
### **Формата Employees**

Във формата Employees се въвеждат данните за заетите на дадена фирма. По структурата и функционалността много прилича на формата Firms. В допълнение има един ComboBox, от който се избира фирмата на служителя от списъка на въведените фирми и PictureBox, където се визуализира снимката на заетия. При въвеждане на нов служител се избира снимка с бутона Select Image (Изберете изображение), от някой локален фолдер. Тази снимка в последствие се записва в базата на MS SQL сървъра в поле от тип Image. Снимките на вече въведените служители също могат да се изменят и ъпдейтват.



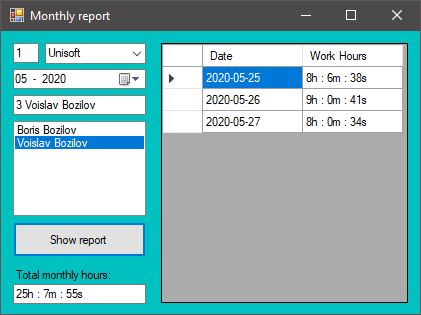
Фигура 5 – Формата на Employees(Служители) с избран български език на интерфейса

### **Формата Attendance**

Формата Attendance служи за визуализация на данните свързани с пристигане и тръгване от работа. В нея има възмошност за избиране на фирмата и датата, за която се отнасят данните. Фирмата се избира от ComboBox, а датата от контрола dateTimePicker, след натискане на бутона Show Attendance от сървъра се зареждат всички записи за служителите от избраната фирма за дадената дата. В dataGridView се зареждат Name, Surname, Last Name и Position от таблицата Employees обединено с Arrival\_Time, Departure\_Time и Work\_Time от таблецата Work\_Attendance. При натискане върху някой от записите в PictureBox-овете се зареждат снимките заснети при пристигане и тръгване от работа, които служат за предотвратяване от злоупотреби с чужд пин(егн) от страна на заетите, т.е. един служител да регистрира повече заети.

Фигура 6 – Формата на Attendance(Посещаемост) на английски и съответно, български език

### **Формата Monthly report**

Формата Monthly report служи за визуализация на данните свързани с работното време на служителите сумарно за всеки ден и общо за целия месец. В нея има възмошност за избиране на фирмата, служителя, годината и месеца. Фирмата се избира от ComboBox, а датата от контрола dateTimePicker, след избор на фирма се визуализират всички служители на фирмата в ListBox, от където може да се избере служителя за който искмаме да извлечем данните. При натискане на бутона Show report от сървъра се зареждат всички записи за даден служител. В dataGridView се зареждат Date и Work hours, от речник, при който колоната Date е ключа, а колоната Work hours представлява стойността в речника.

Фигура 7 - Формата на Monthly report на английски език

В продължение, **за илюстрация** е даден кодът на метода btnShowReport, който тегли данните от базата с SQL-заявка, запълва речника с дании и върши обработка (събиране) на работното време в часове, минути и секунди за всеки работен ден, а след това създава нов речник който се използва за визулализация на данните, чрез dataGridView:

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

{

string[] employee = employeeTextBox.Text.ToString().Split(' ');

Dictionary<int, List<string>> selectTableLanguage = new Dictionary<int, List<string>>

{

{ 0, new List<string> { "Date", "Work Hours" } },

{ 1, new List<string> { "Дата", "Работни часове" } },

{ 2, new List<string> { "Датум", "Радно време" } }

};

myConnection.Open();

//.ToString("yyyy-MM-dd")

SqlDataAdapter myAdapter = new SqlDataAdapter("select Date\_att, (SUM((DATEPART(hh,Work\_Time)\*60)+DATEPART(mi,Work\_Time)+(DATEPART(ss,Work\_Time)/(60.0)))\*60) as Work\_Time from Work\_Attendance where (Work\_Attendance.Firm\_ID = " + firmTextBox.Text.ToString() + " and Work\_Attendance.Employee\_ID = " + employee[0] + " and (SELECT MONTH(Work\_Attendance.Date\_Att) AS Month) = '" + dateTimePicker.Value.Month + "') GROUP BY Date\_att", myConnection);

DataTable dtMonthlyReport = new DataTable();

myAdapter.Fill(dtMonthlyReport);

Dictionary<DateTime, decimal> dictionaryMonthlyReport = GetDict(dtMonthlyReport);

Dictionary<DateTime, string> dictionaryStringMonthlyReport = new Dictionary<DateTime, string>();

//monthlyReportDataGridView.DataSource = dtMonthlyReport;

decimal sumTime = 0;

decimal hours = 0;

decimal minutes = 0;

decimal seconds = 0;

foreach (var item in dictionaryMonthlyReport)

{

hours = (int)(item.Value / 3600);

minutes = (int)((item.Value % 3600) / 60);

seconds = (int)((item.Value % 3600) % 60);

dictionaryStringMonthlyReport.Add(item.Key, $"{hours}h : {minutes}m : {seconds}s");

}

monthlyReportDataGridView.DataSource = (from d in dictionaryStringMonthlyReport orderby d.Key select new { d.Key, d.Value }).ToList();

var value = selectTableLanguage[Program.LanguageKey];

monthlyReportDataGridView.Columns[0].HeaderText = value[0];

monthlyReportDataGridView.Columns[1].HeaderText = value[1];

foreach (var item in dictionaryMonthlyReport)

{

sumTime += item.Value;

}

hours = (int)(sumTime / 3600);

minutes = (int)((sumTime % 3600) / 60);

seconds = (int)((sumTime % 3600) % 60);

monthlyReportTextBox.Text = hours.ToString() + "h : " + minutes.ToString() + "m : " + seconds.ToString() + "s";

myConnection.Close();

}

private Dictionary<DateTime, decimal> GetDict(DataTable dt)

{

return dt.AsEnumerable()

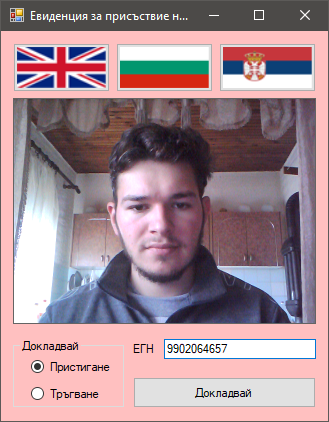
.ToDictionary<DataRow, DateTime, decimal>(row => row.Field<DateTime>(0),

row => row.Field<decimal>(1));

}

## **Приложението Work Attendance Evidence**

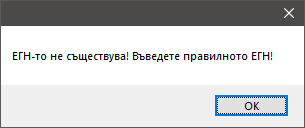
Приложението се използва за отчитане на пристигането и тръгването от работа, което е предназначено да работи на отделен компютър, намиращ се на входната врата на сградата. То има само една форма с два радио бутона, с които може да се избере дали служителя пристига или си тръгва от работа, поле за егн на служителя, с помощта на което той се идентифицира, PictureBox за видео наблюдение и бутон за заявка на пристигане или тръгване. Също така, като и в първото приложение са добавени във върха на формата са расположени бутоните за избор на езика на приложението.



Фигура 8 – Форма на приложението Work Attendance Evidence

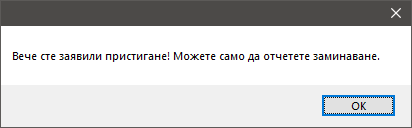
(Евиденция на присъствие на персонала) на български език

При натискане на бутона Report(Докладвай) приложението проверява дали съществува служител с въведения пин. Ако не съществува му извежда съобщение:



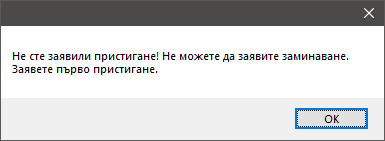
Фигура 9 – MessageBox със съобщение за грешно въведен пин

В случай, че съществува служител с въведения пин, проверява дали е избрал Arrival(Пристигане) или Departure(Тръгване). Ако е избрал Arrival проверява дали този служител не е предварително заявил пристигане. В случай че е, излиза съобщение, че вече е заявил пристигане.



Фигура 10 – MassageBox за вече заявено пристигане

Подобна логика има и в заявката за тръгване от работа.



Фигура 11 – MassageBox за не заявено пристигане

Системата позволява повече пристигания и напускания в един ден, заради нужда за излизане от работа поради обедна почивка и пр.

Програмата лесно може да се модифицира за въвеждането на ЕГН-то, с помощта на бар код или QR-код четец и идентификационни карти на служителите, което премахва нуждата от въвеждането на ЕГН-то от клавиатурата. По нататък може да се интегрира с разни актуатори за отваряне на вратата и подобно.

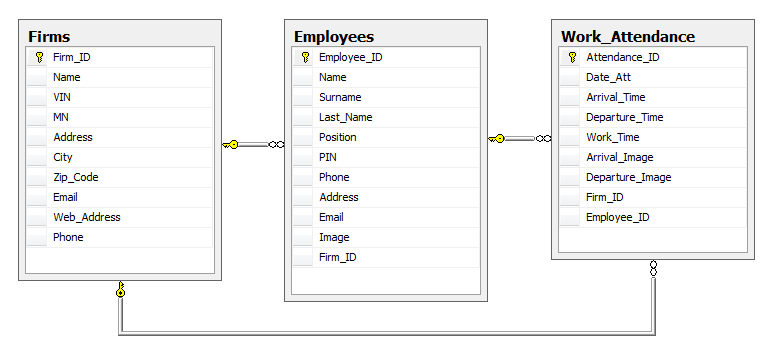
# **Описание на базата данни**

При проектирането на базата данни е важно да се включат всички данни, които са необходими за изпълнението на поставените задачи, а също така да се избягват ненужните. За да не възникнат проблеми на по-късен етап е необходимо да се използва процесът на нормализация. Нормализацията изпълнява следните задачи:

* Изключване на повтаряща се информация в таблиците;
* Декомпозиция на смесените полета;
* Избиране на полетата така да са в зависимост само от първичния ключ и да не зависят едно от друго

В настоящата база данни е направено едно изключение от правилата за нормализация при полето Work\_time което се изчислява като разлика на Departure\_time и Arrival\_time. Изключението е направено поради облекчение за събирането на работното време при създаването на отчетите на по късен етап.

В следващата фигура са показани релациите между използваните таблици, като и структурите на самите таблици.



Фигура 12 – Диаграма на базата данни

# **Инсталация на системата**

За да работи правилно приложението трябва да се направят няколко предварителни стъпки:

1. Трябва да се отбележи, че приложенията са създадени във Visual Studio - версия 2019 и базата данни е създадена в MS SQL Server – версия 2019.
2. В инстанция на MS SQL Server трябва да се създаде нова база данни с название Time\_Evidence\_2020 oт файла с бекъпа на базата данни Time\_Evidence\_2020.bak с командата **Tasks/Restore/Database…**, който е приложен в папка: SDA-46261r\SDA\_46261r\_MyProject.
3. Трябва да се смени connection string-а за двете приложения на системата в файла app.config, който се намира в фолдерите на приложенията, така че да сочи на правилната инстанция на SQL сървъра.
4. В приложението Work Attendance Evidence, в неговата bin/Debug директория трябва да се добави файла WebCam\_Capture.dll, който е необходим за работа на уеб камерата т.е. без който приложението не може да се стартира.

# **Заключение**

С развитието на съвременните технически средства за електронно отчитане и наблюдение, контролът, който може да упражнява работодателят над работниците и служителите в дадено предприятие като част от мерките за гарантиране на трудовата дисциплина, в последните години придобива все по-голямо значение. Използването на система за контрол на работното време вече се налага като необходимист за всяка организация и бизнес.

Създаденото приложение, не претендира да реши тази комплексна проблематика, но може да се доразвива в посока интеграция с приложения за контрол на достъп, приложения за изчисляване на трудови възнаграждения и пр.

Целта на разработенето приложение е да покаже основните компоненти на една такава система и да послужи за демонстрация на знанията и уменията за разработка на приложения в съвременните визуални среди за програмиране, както ипо-специално **умения за употреба на някои от структурите от данни заложени в езика C#**, в конкретния случай **речници реализирани с хеш таблици, списъци и масиви**.

За всяка конкретна задача от ключово значение е изборът на правилната структура. Това спестява памет и време за изпълнение, което прави програмата много по-ефективна.