**УНИВЕРСИТЕТ ПО БИБЛИОТЕКОЗНАНИЕ И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ**

**ФАКУЛТЕТ „ИНФОРМАЦИОННИ НАУКИ“**

**К У Р С О В А Р А Б О Т А**

по

СТРУКТУРИ ОТ ДАННИ И АЛГОРИТМИ

**ТЕМА**: **„Използване на структури от данни и алгоритми за създаване на приложение за паркинг система за въвеждане на данни на клиент, ППС и изчисляване на очаквани приходи“**

Изготвил: Преподавател:

Росица Ненова проф. д-р Иван Иванов

ФН 46231з

София

2021

Съдържание

[Увод 3](#_Toc68525822)

[I. Функции на приложението 3](#_Toc68525823)

[1. Tab „Въвеждане на ППС“ 3](#_Toc68525824)

[2. Tab „Финанси“ 4](#_Toc68525825)

[3. Tab „Ценоразпис“ 5](#_Toc68525826)

[II. Създаване на дизайна 6](#_Toc68525827)

[1. Създаване на ново *Windows* *приложение* 6](#_Toc68525828)

[2. Вмъкване на контролите *TabControl* 6](#_Toc68525829)

[3. Вмъкване на контролите в tab „Въвеждане на ППС“ 7](#_Toc68525830)

[4. Вмъкване на контролите в tab „Финанси“ 8](#_Toc68525831)

[5. Вмъкване на контролите в tab „Ценоразпис“ 8](#_Toc68525832)

[III. Създаване на функционалностите 9](#_Toc68525833)

[1. Методи в tab „Въвеждане на ППС“ 9](#_Toc68525834)

[2. Методи в таб „Финанси“ 18](#_Toc68525835)

[3. Метод в таб „Ценоразпис“ 25](#_Toc68525836)

[Библиография 28](#_Toc68525837)

# Увод

В настоящата курсова работа ще разгледаме последователните стъпки и използваните контроли и код за създаване на приложение за въвеждане на данни за клиент (собственик на ППС) и неговото (неговите) ППС относно запазване на паркомясто, както и изчисляване на очаквани доходи за месеца спрямо въведените данни. Потребителският интерфейс ще бъде показан на следващите страници.

## Функции на приложението

Приложението се състои от три таба, всеки от които има отделна функция.

### Tab „Въвеждане на ППС“

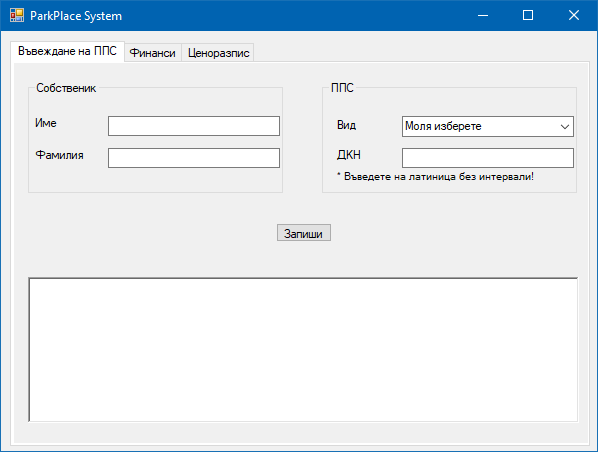
В първия таб с име „Въвеждане на ППС“ има няколко полета за въвеждане на данни. В текстови полета, отделени в специален GroupBox, се въвеждат първо и фамилно име на собственика на ППС, който е клиент.

Във втори GroupBox се въвежда информация за пътното-превозно средство, за което ще се запазва паркомясто. Информацията за ППС-то е два вида – тип на ППС и регистрационен номер. От падащ списък се избира типа на ППС, който определя и ценообразуването. В текстово поле се въвежда регистрационния номер, а под въпросното текството поле има информация какъв е коректният начин за попълване на тези данни.

С бутон „Запиши“ се записват данните на собственика и на ППС.

В текстово поле се визуализира потвърждение, че е запазено парко място за ППС със съотвения регистрационен номер и собственик.

Изгледът на интерфейса може да се види на Фигура 1.



Фигура Интерфейс на tab „Въвеждане на ППС“

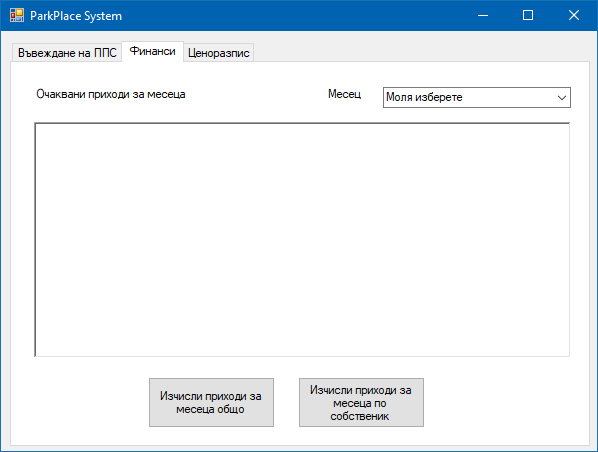
### Tab „Финанси“

Във втория таб с име „Финанси“ се визуализират очакваните приходи за месеца.

С падащ списък се избира месеца, за който ще се изчисляват очакваните приходи, а самите суми се визуализират в текстово поле.

Изчислението на сумите се прави посредством два бутона, които дават възможност за избор между визуализиране на очакваните приходи общо по регистрационен номер и визуализиране на очакваните приходи общо по собственик на ППС, тъй като има възможност един и същ собственик да е абониран за две и повече паркоместа.

Изгледът на интерфейса може да се види на Фигура 2.



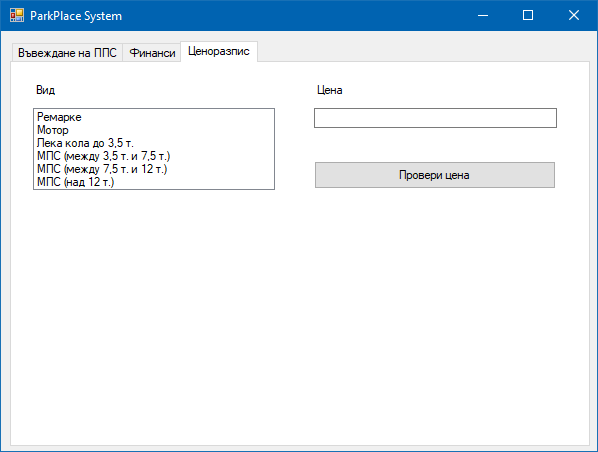
Фигура Интерфейс на tab „Финанси“

### Tab „Ценоразпис“

Приложението предвижда възможност потребителят му да не знае всички цени наизуст и за тези случаи е разработен и tab „Ценоразпис“.

При използването му е нужно само да се маркира дадения тип ППС, след което при натискане на бутон „Провери цена“ в текстово поле се визуализира цената, съответстваща на избрания вид ППС.

Изгледът на интерфейса може да се види на Фигура 3.



Фигура Интерфейс на tab „Ценоразпис“

## Създаване на дизайна

Създаването на изгледа на приложението се изпълнява в няколко стъпки.

### Създаване на ново *Windows* *приложение*

Създава се ново *Windows* *приложение* с наименование SDA\_46231z\_10. Размерите на формата е препоръчително да са „812, 564“. Минималният размер и максималният размер получават същите стойности. Стойността на свойството *Name* е „ParkPlace System“.

### Вмъкване на контролите *TabControl*

Вмъква се контрола от типа *TabControl*, като в свойството *TabPages* се въвеждат имената на трите таб контроли, които ще се използват.

|  |  |  |
| --- | --- | --- |
| Tab | Name | Text |
| 1 | tabAddVehicle | Въвеждане на ППС |
| 2 | tabFinance | Финанси |
| 3 | tabPriceList | Ценоразпис |

Таблица Свойства на Tab констролите

### Вмъкване на контролите в tab „Въвеждане на ППС“

Вмъкват се няколко контроли. В Таблица 2 са показани стойностите, които трябва да се присвоят на *Name* и *Text* свойствата, както и типовете на самите контроли.

|  |  |  |
| --- | --- | --- |
| Type | Name | Text |
| groupBox | gbAddVehicleOwner | Собственик |
| groupBox | gbAddVehiclePPS | ППС |
| Button | btnAddVehicle | Запиши |
| richTextBox | rtbAddVehicleInfo | - |
| label | lblAddVehicleOwnerNameFirst | Име |
| label | lblAddVehicleOwnerNameLast | Фамилия |
| label | lblAddVehiclePPSType | Вид |
| label | lblAddVehiclePPSDKN | ДКН |
| label | lblAddVehicleInfo | \* Въведете на латиница без интервали! |
| textBox | tbAddVehicleOwnerNameFirst | - |
| textBox | tbAddVehicleOwnerNameLast | - |
| textBox | tbAddVehiclePPSDKN | - |
| comboBox | cbAddVehiclePPSType | Моля изберете |

Таблица Свойства на контролите от tab „Въвеждане на ППС“

Важно е да се отбележи за *comboBox* контролата, че чрез свойството *Items* е зададена колекция от видовете ППС, които могат да бъдат избирани.

### Вмъкване на контролите в tab „Финанси“

Вмъкват се няколко контроли. В Таблица 3 са показани стойностите, които трябва да се присвоят на *Name* и *Text* свойствата, както и типовете на самите контроли.

|  |  |  |
| --- | --- | --- |
| Type | Name | Text |
| label | lblFinance | Очаквани приходи за месеца |
| label | lblFinanceMonth | Месец |
| comboBox | cbFinanceMonth | Моля изберете |
| richTextBox | rtbFinanceResult | - |
| button | btnFinanceCalculateAll | Изчисли приходи за месеца общо |
| button | btnFinanceCalculateOwner | Изчисли приходи за месеца по собственик |

Таблица Свойства на контролите от tab „Финанси“

Важно е да се отбележи за *comboBox* контролата, че чрез свойството *Items* е зададена колекция от месеците през годината.

### Вмъкване на контролите в tab „Ценоразпис“

Вмъкват се няколко контроли. В Таблица 4 са показани стойностите, които трябва да се присвоят на *Name* и *Text* свойствата, както и типовете на самите контроли.

|  |  |  |
| --- | --- | --- |
| Type | Name | Text |
| listBox | lbPricelistType | - |
| label | lblPricelistType | Вид |
| label | lblPricelistPrice | Цена |
| textbox | tbPricelistPrice | - |
| button | btnPricelistCheck | Провери цена |

Таблица Свойства на контролите от tab „Ценоразпис“

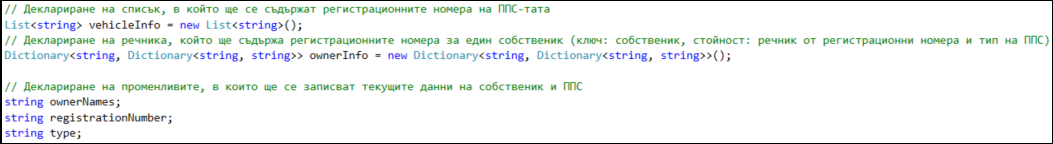
Важно е да се отбележи за *listBox* контролата, че чрез свойството *Items* е зададена колекция от възможните типове ППС.

## Създаване на функционалностите

### Методи в tab „Въвеждане на ППС“

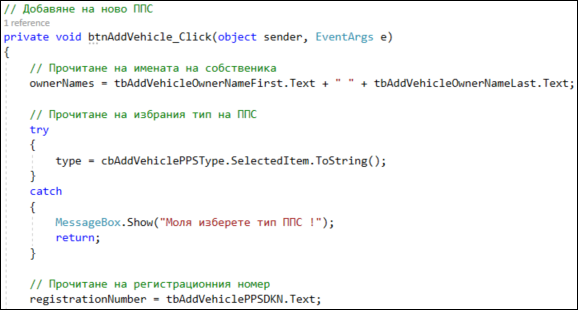
В tab „Въвеждане на ППС“ има имплементирани няколко метода.

В началото на програмата декларираме структурите от данни, които ще изпозлваме за целите на приложението. На този етап това са списък, в който ще се съдържат въведените регистрационни номера, както и два речника, като единият от тях е вложен в другия като свойство на ключа на първия. На Фигура 4 е показано декларирането на съответните структури от данни. На фигурата също така присъства и декларирането на помощните променливи, в които ще записваме въведените стойности от потребителя. Чрез тези променливи ще записване данните в гореспоменатите речници и списък.



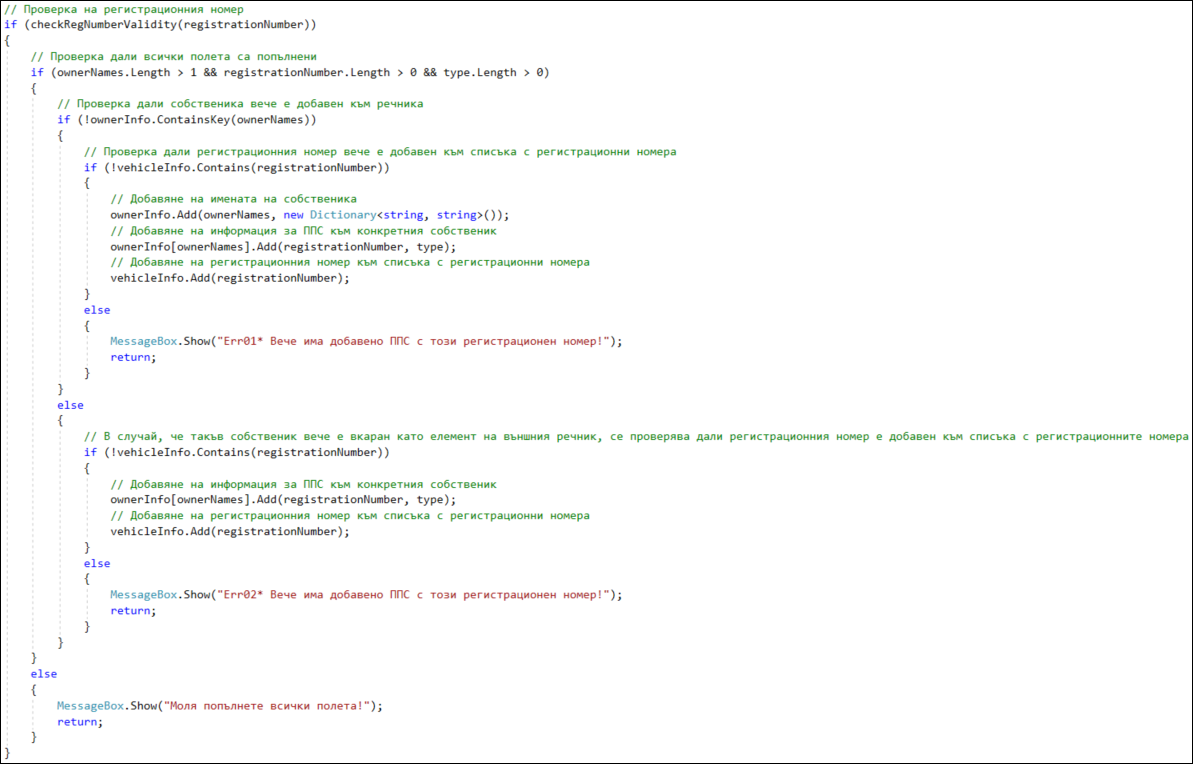
Фигура Деклариране на структури и типове данни

След декларирането на структурите от данни и променливите, които ще се използват на този етап, следва самия метод за записване на данни на собственик и ППС, който се извиква при натиска на бутона с *Name* „btnAddVehicle“. В началната част от кода се прочитат въведените данни от потребителя и стойностите им се запазват във вече декларираните променливи (Фигура 5).



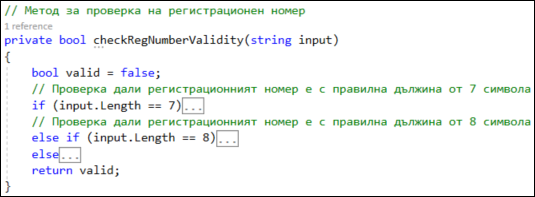
Фигура Прочитане на стойности, въведени от потребител

След прочитането на входните данни, въведени от потребителя, в метода има имплементирани няколко проверки относно коректността на данните. Тези проверки могат да бъдат разгледани на Фигура 6.



Фигура Проверки на входни данни

Първо се проверява дали регистрационният номер е въведен според стандартите на Република България. Проверката се извършва чрез извикване на отделен метод, който може да бъде разгледан на Фигура 7.

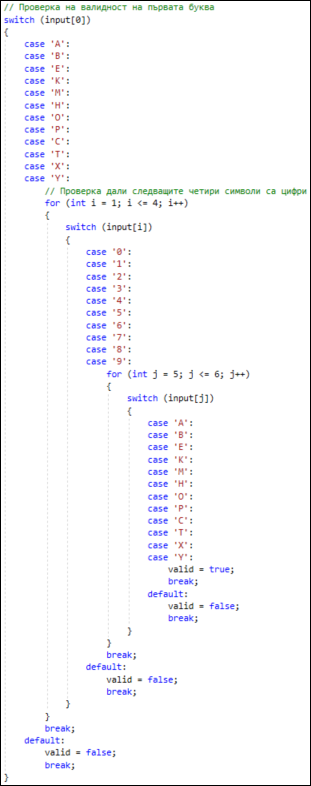


Фигура Проверка на валидност на ДКН на ППС

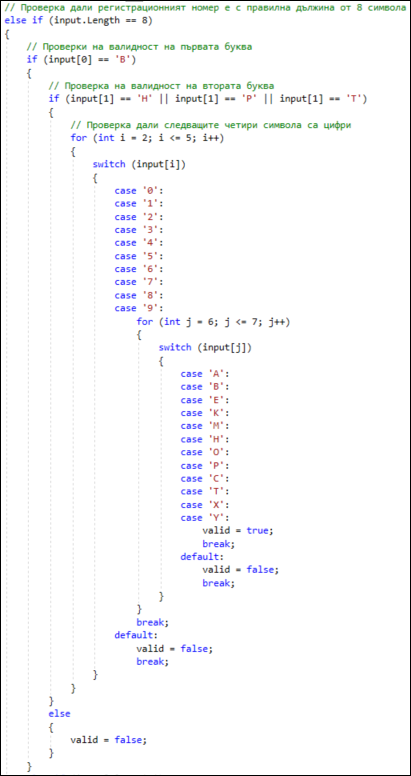
В следващите редове ще разгледаме подробно и метода за проверка на валидност на ДКН на съответното ППС.

Първо се проверява дължината на регистрационния номер. За България са валидни две дължини – една от 7 символа и една от 8 символа. В случай, че въведеният ДКН има дължина от 7 символа, то се проверява дали първия от тях е един от възможните. В случай, че първият символ отговаря на правилата за валиден ДКН, то се проверява дали следващите символи са цифри. Щом и тази проверка върне резултат true, то се проверява дали последните два символа са някои от позволените букви, които могат да участват във формирането на регистрационен номер (Фигура 8).

Подобна проверка се прави и за въведните ДКН с дължина 8 символа, като разликата тук е, че се прави допълнителна проверка за втория символ дали съответства на някоя от разрешените букви за формиране на български регистрационен номер (Фигура 9).

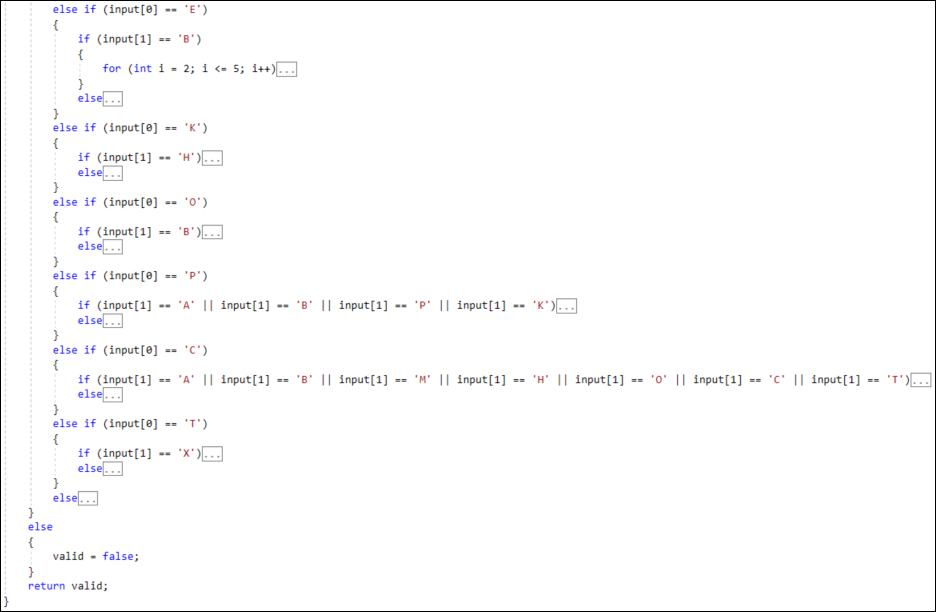


Фигура Проверка на въведените ДКН с дължина 7 символа



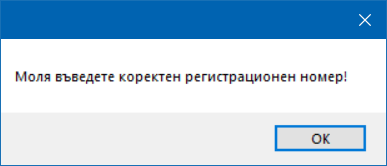
Фигура Проверка на въведените ДКН с дължина 8 символа

Аналогична проверка се прави за останалите възможни комбинации от позволени букви за формирането на валиден български ДКН (Фигура 10).



Фигура Проверка на останалите възможни комбинации

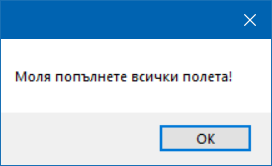
Ако методът за проверка на валидността на регистрационния номер върне стойност false, то излиза съобщение, в което се адресира потребителя и му се указва да въведе коректно регистрационния номер на съответното ППС (Фигура 11).



Фигура Съобщение за некоректно въведен регистрационен номер

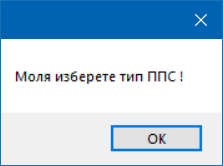
В случай, че гореспоменатият метод върне стойност true, то програмата продължава към следващата проверка, а именно дали всички полета са попълнени. Проверката се извършва като се проверява дали дължината на въведените стойности е по-голяма от 0 (нула). Изключение прави дължината на името на собственика, тъй като минималната му дължина е 1, заради интервала между първото и фамилното име.

Ако тази проверка върне стойност false, то програмата отново визуализира прозорец, който указва на потребителя да попълни всички полета (Фигура 12).



Фигура Съобщение за попълване на всички полета

Необходимо е да се спомене, че още при четенето на данните от попълнените текстови полета е използвана конструкцията try-catch за прочитане на избрания тип на ППС, тъй като в случай, че не е избран тип ППС, то програмата ще хвърли грешка. Резултатът от изпълнението на try-catch в случай на неизбран тип на ППС може да се види на Фигура 13.



Фигура Резултат от try-catch

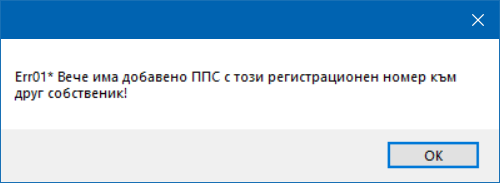
В случай, че са минали всички проверки и резултатите от тях позволяват на програмата да продължи, то се стига до частта на метода, в която се работи със структурите от данни.

Първоначално се проверява дали външният речник с елементи с ключ имената на собственика съдържат текущите въведени имена от потребителя. В случай, че речникът все още не съдържа елемент с такъв ключ, то се добавя този елемент, който ще бъде външния речник, а като стойност се задава нов речник с елементи с ключ string и стойност string, който се явява вътрешния речник.

При този случай се въвеждат като ключ и стойност регистрационния номер и типа на ППС.

Допълнително се добавя и регистрационния номер към списъка с регистрационни номера, за да може да се прави последваща проверка дали вече е въведен такъв ДКН.

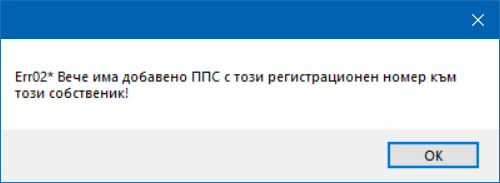
Ако проверката по-горе установи, че вече има въведено ППС с такъв ДКН, а няма въведен такъв собственик, то се извежда съобщение, което предупреждава потребителя за това и дава възможност да се коригират данните (Фигура 14).



Фигура Съобщение за въведено ППС към друг собственик

В случай, че проверката по-горе установи, че вече има въведен такъв собственик като ключ на елемент от външния речник, то програмата проверява дали вече има въведено ППС с такъв регистрационен номер. Ако проверката установи, че няма въведен такъв регистрационен номер към списъка с регистрационни номера, то във вътрешния речник към съответния собственик се въвеждат стойностите на регистрационния номер и типа на ППС.

В случай, че проверката установи, че в списъка с регистрационни номера фигурира това ДКН, то се извежда съобщение, което предупреждава потребителя за това и се дава възможност да се коригират данните (Фигура 15).



Фигура Съобщение с предупреждение, че е въведено такова ППС към този собственик

В края на метода в текстово поле се визуализира потвърждение, че са въведени данните съсм самите тях.

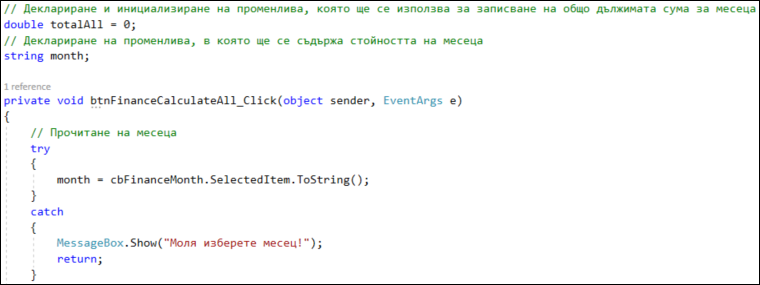
### Методи в таб „Финанси“

В таб „Финанси“ отново има имплементирани няколко метода.

Първият, който ще разгледаме, е този, който изчислява по регистрационен номер очакваните приходи за даден месец на база въведените данни за собственици и типове ППС-та. Той е зададен като Property при event Click върху бутон btnFinanceCalculateAll.

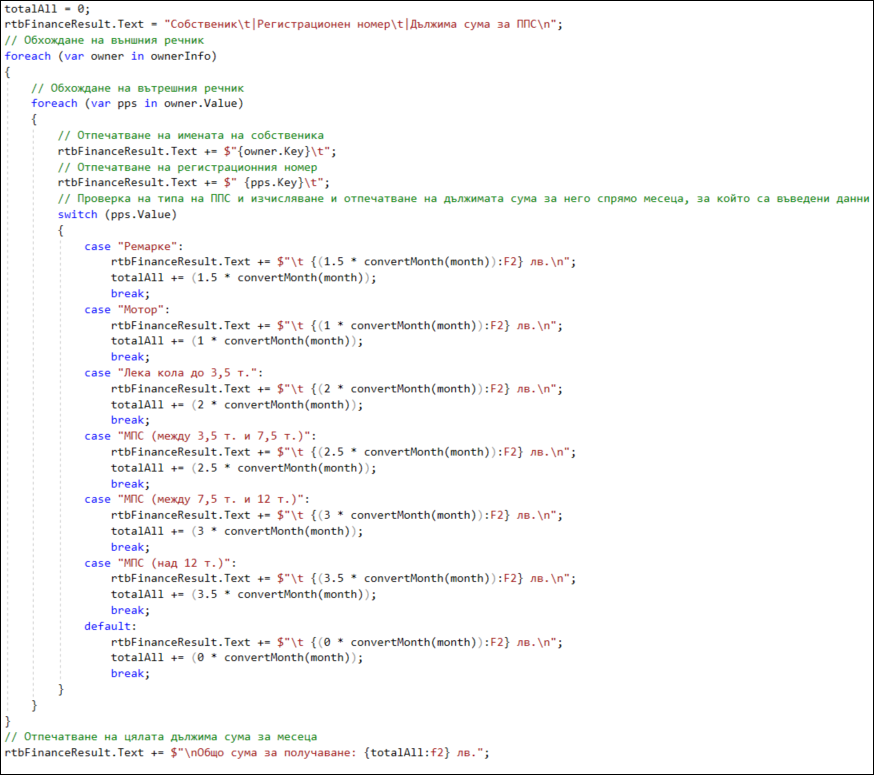
Преди него обаче е наложително да се декларират две променливи – едната ще е нужна при записване на общите дължими суми за месеца, а другата ще е нужна за изчисляване на отделните месечни суми.

Тъй като тук отново се работи с падащ списък, е наложително да се използва конструкцията try-catch, която ще предотврати счупването на програмата, в случай, че потребителят е забравил да избере месец (Фигура 16).



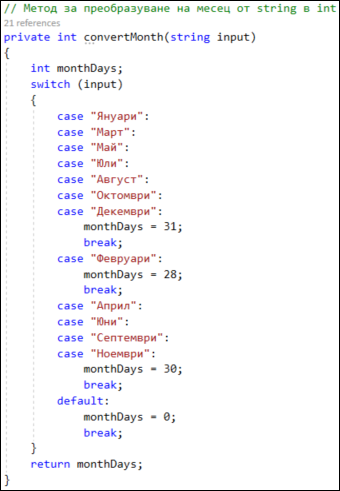
Фигура Try-catch конструкция при четене на стойност на месец

След като се въведe месеца, първо се прави обхождане на външния речник, т.е. елементите, които са ключове към други речници. След това техните стойности, бидейки вътрешния речник, биват обходени също, за да се сравнят стойностите на елементите от вътрешния речник със типовете ППС. При съвпадение на данните, се изчислява месечната дължима сума, а в помощната променлива се натрупва общата дължима сума до момента (Фигура 17).



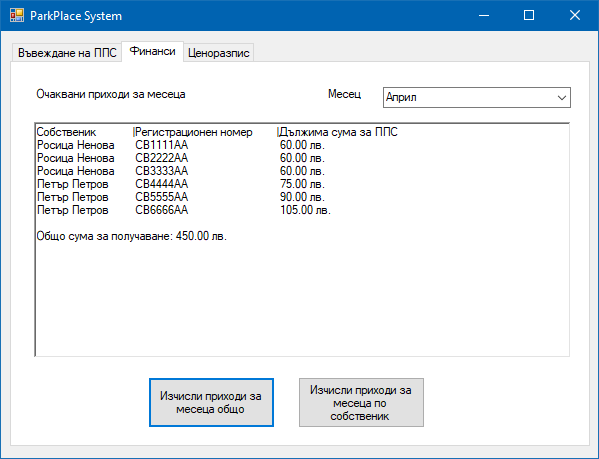
Фигура Изчисление на дължима сума по регистрационен номер

При изчислението на месечната сума за всяко едно ППС отделно, се използва допълнителен метод, чрез който името на месеца се конвертира в число, представляващо броя на дните, които съответстват на избрания месец (Фигура 18).



Фигура Метод за конвертиране

При коректно изпълнение на програмата, в текстовото поле се визуализират данни за собственик, регистрационен номер на ППС, както и дължимата сума за съответното ППС (Фигура 19).



Фигура Визуализиране на резултат

По подобен начин работи и методът за изчисляване на очаквани приходи по собственик. Има малки разлики, които ще разгледаме в следващите редове (Фигура 20).



Фигура Метод за изчисление на суми по собственик

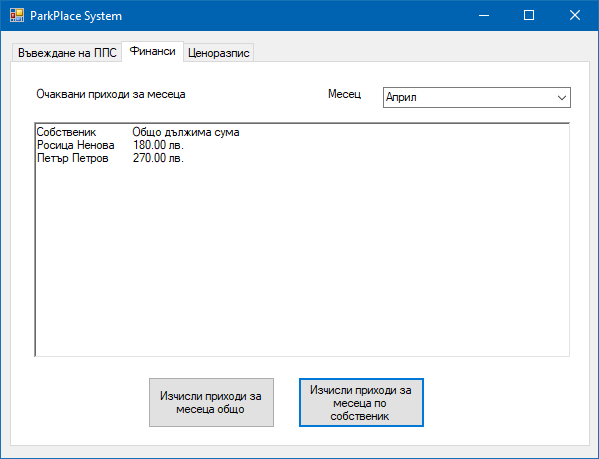
Декларирането на променлива, в която ще се натрупват отделните стойности по собственик, е наложително. Декларира се и нов речник, в който като ключове ще присъстват имената на собствениците, а стойности ще бъдат сумите за всеки един собственик отделно.

След това първо се обхожда външния речник с елементи, чиито ключове са имената на собствениците, а стойностите им са вътрешните речници, които съдържат елементи с ключове регистрационните номера и стойности типа на ППС.

Прави се проверка дали името на собственика от външния речник съвпада с името на собственик от новия речник. В случай, че не е открит такъв елемент, то се създава като за ключ се задават имената на съответния собственик, а за стойност се задава променливата, в която ще се съдържат сумите по собственици.

След това се обхожда и вътрешния речник, за да е възможна проверката на типовете на ППС-тата, които са записани към дадения собственик. Проверявайки типа им и използвайки отново метода за конвертиране на името на месеца към брой дни, се изчислява дължимата сума по собственик и се записва в новия речник.

При коректно изпълнение на програмата се разпечатва списък с имена на собственици и общите дължими суми от тях. Разпечатването става като се обходжа с цикъл новия речник и се разпечатват ключовете и стойностите на всеки един елемент (Фигура 21).



Фигура Метод за отпечатване на суми по собственици

### Метод в таб „Ценоразпис“

При създаване на приложението е предвидено, че потребителят може да не знае наизуст всички цени, зависещи от типа на ППС. За тези случаи е разработен последният таб „Ценоразпис“.

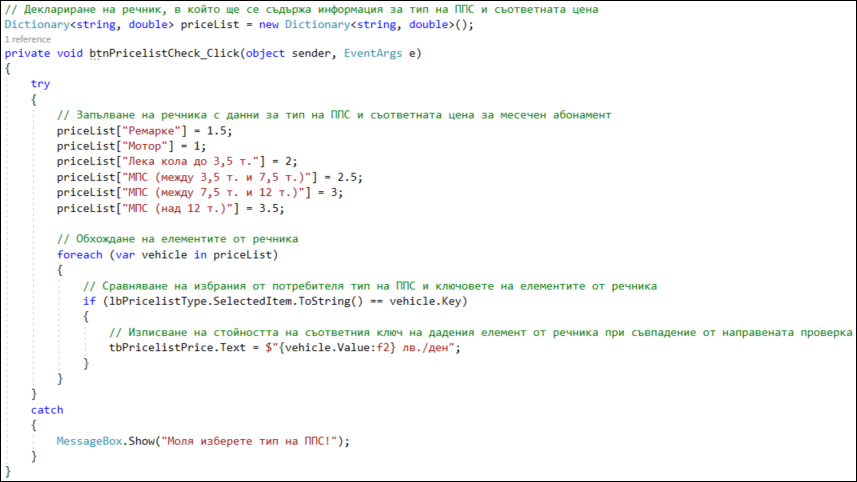
От списък се избира типа ППС, за който се интересува потребителят, и се натиска бутон „Провери цена“. Към този бутон е закачен метод, който визуализира каква е цената за съответния тип ППС.

Това е реализирано чрез речник, чийто елементи имат ключ тип на ППС и стойности – цената на ден за съответния тип ППС.

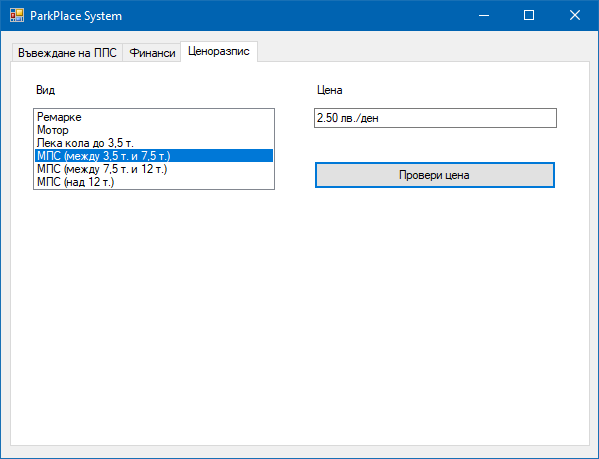
Речникът се обхожда и се сверяват ключовете на елементите спрямо данните, които е избрал потребителят. При намерено съвпадение в текстово поле се визуализира цената на ден, която се явява стойност на елемента от речника.

Отново е използвана конструкция try-catch, която да предотврати счупването на програмата, в случай, че потребителят забрави да избере тип на ППС и натисне бутона за изчисление преди това.

Кодът на метода може да бъде разгледан на Фигура 22, а резултатът може да бъде видян на Фигура 23.



Фигура Метод за визуализиране на цена



Фигура Резултат от метод за визуализиране на цена

# Библиография

Структури от данни и алгоритми, проф. д-р Иван Иванов, доц. д-р Петър Стойков, Про Лангс, София, 2015

Принципи на програмирането със C#, д-р Светлин Наков, Веселин Колев и колектив, Софтуерен Университет, София, 2018

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

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