Частное учреждение образования

«Колледж бизнеса и права»

Программное средство для расчета страховых накопительных выплат

Пояснительная записка

к курсовому проекту по учебному предмету «Основы алгоритмизации и программирования»

КП Т.219019.401

Руководитель проекта (Ю.В. Шаляпин)

Обучающаяся ( А.В. Писарик )

2023

**СОДЕРЖАНИЕ**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

КП Т.219019.401

Разраб.

Писарик А.В.

Провер.

Шаляпин Ю.В.

Реценз.

Н. Контр.

Утверд.

*Программное средство для расчета страховых накопительных выплат*

Лит.

Листов

КБП

63

у

[Введение 4](#_Toc138855668)

[1 Объектно-ориентированный анализ и проектирование приложения 6](#_Toc138855669)

[1.1 Сущность задачи 6](#_Toc138855670)

[1.2 Проектирование модели 6](#_Toc138855671)

[2 Вычислительная система 8](#_Toc138855672)

[2.1 Требования к аппаратным и операционным ресурсам 8](#_Toc138855673)

[2.2 Инструменты разработки 8](#_Toc138855674)

[3 Проектирование задачи 9](#_Toc138855675)

[3.1 Требования к приложению 9](#_Toc138855676)

[3.2 Концептуальный прототип 9](#_Toc138855677)

[3.3 Функции и элементы управления 12](#_Toc138855678)

[3.4 Проектирование справочной системы приложения 17](#_Toc138855679)

[4 Описание программного средства 18](#_Toc138855680)

[4.1 Общие сведения 18](#_Toc138855681)

[4.2 Входные и выходные данные 18](#_Toc138855682)

[5 Методика испытаний 19](#_Toc138855683)

[5.1 Технические требования 19](#_Toc138855684)

[5.2 Функциональное тестирование 19](#_Toc138855685)

[6 Применение 22](#_Toc138855686)

[6.1 Назначение программы 22](#_Toc138855687)

[6.2 Условия применения 22](#_Toc138855688)

[6.3 Справочная система 22](#_Toc138855689)

[Заключение 24](#_Toc138855690)

[Список информационных источников 25](#_Toc138855691)

[ПРИЛОЖЕНИЕ А Текст программы 26](#_Toc138855692)

[ПРИЛОЖЕНИЕ Б Результат работы программы 52](#_Toc138855693)

# Введение

Автоматизация повышает деятельность предприятия путем внедрения технического оснащения и соответствующего софта, в результате чего значительная часть работы может проводиться с минимальным количеством персонала. Это позволяет избегать ошибок и задержек, обуславливаемых человеческим фактором. Основной инструмент автоматизации – компьютерная система, разделенная на секторы и способная выполнять множество процессов одновременно.

Целью курсового проекта для автоматизации расчета накопительного страхования является создание проекта «курсовая\_2курс.exe», осуществляющий расчет уплаты страхового взноса по накопительному страхованию исходя из выбранного тарифа, сохранённых данных о клиенте и лимите ответственности по страховому продукту.

Пояснительная записка к курсовому проекту состоит из шести разделов, содержащих необходимую информацию по организации эксплуатации программного приложения.

В первом разделе «Объектно–ориентированный анализ системы и проектирование системы» раскрывается организационная сущность задачи, описывается предметная область и круг задач, которые должны быть автоматизированы. Описывается задача, перечисляются основные функции программы. Строится информационная модель, отражающая сущности задачи, их свойства и взаимосвязи.

Во втором разделе «Вычислительная система» перечисляются требования к аппаратному обеспечению и конфигурации компьютера, проводится характеристика операционной системы, обоснование выбранной среды для разработки приложения.

В третьем разделе «Проектирование задачи» указываются требования к приложению (средства защиты, требования к интерфейсу), строится концептуальный прототип системы меню, диалоговых окон и элементов управления, приводится организация данных, функций и элементов управления, задействованных в программном средстве.

В четвертом разделе «Описание программного средства» представлены общие сведения о программном средстве, обозначение и наименование приложения, программное обеспечение, необходимое для функционирования системы, способ инсталляции программного средства, средства защиты, приводится структура входных и выходных данных.

В пятом разделе «Методика испытаний» описываются требования к техническим средствам для проведения испытаний, требования к характеристикам программы применительно к условиям эксплуатации, требования к информационной и программной совместимости. Представляются результаты функционального тестирования.

Шестой раздел «Применение» предназначен для описания сведений о назначении программного средства и области его применения.

В заключении будет проанализировано созданное программное средство, определена степень соответствия поставленной задачи и выполненной работы.

Приложение А будет содержать текст программы.

Приложение Б будет результат программы.

Графическая часть представлена диаграммой алгоритма расчета периодического взноса накопительной страховки и диаграммы работы программы.

# Объектно-ориентированный анализ и проектирование приложения

## Сущность задачи

Требуется разработать программное средство для расчета накопительных страховых выплат. Следовательно, эта программа для расчета накопительного страхования и вычисления выплат в течении срока действия договора.

Предметной областью данной программы может являться любая страховая компания, которая предоставляет программу накопительного страхования.

Калькулятор накопительной страховки человека – это инструмент, который помогает рассчитать сумму накопления по программе накопительного страхования, узнать сумму выплаты при наступлении страхового случая или окончания программы.

Данная программа может автоматизировать и упростить процесс подсчета накопительных уплат исходя из тарифов, периодичности уплат и взносов по страховым случаям.

Основной аудиторией являются работники страховых компаний, а также их клиенты.

Исходя из этого, можно выделить следующие задачи для программного продукта, а именно:

* программа должна быть понятна конечному пользователю;
* программа должна легко и быстро настраиваться;
* программа должна иметь 3 разных темы оформления, это обусловлено тем, что офисы работают как в вечернюю смену, так и в дневную;
* программа должна поддерживать английский, русский и белорусский язык.

Аналогов данного программного средства не существует, т.к. подобных программных средств с открытым кодом не существует, либо разрабатывались другими предприятиями не публично.

## Проектирование модели

Блок-схема – это графическое представление последовательности операций или алгоритма с использованием различных символов и компонентов.

В блок схемах используются следующие символы и компоненты для создания блок-схемы:

* прямоугольники для представления операций или шагов в вашем алгоритме. Пишется внутри каждого прямоугольника название операции;
* ромбы для условных операторов, таких как «if» или «else». Пишется внутри ромба условие, которое должно быть проверено;
* стрелки для соединения операций и показа последовательности выполнения. Указываются направление стрелки, чтобы показать, какие операции следуют за какими;
* параллельные линии используются если несколько операций, которые должны выполняться параллельно, используются параллельные линии для отображения этого;
* ввод/вывод если алгоритм включает операции ввода или вывода данных, используются символы, такие как стрелки вправо или влево, чтобы показать это.

Блок-схема должна быть понятной и логически связанной, чтобы люди могли легко понять ваш алгоритм или последовательность операций. Исходя из этого были созданы следующие блок схемы:

* блок-схема работы программы;
* блок-схема работы одного из алгоритмов расчёта производной неявной функции.

Перечисленные выше блок-схемы представлены в графической части, блок-схема работы программы представлена на листе 1 графической части, а блок-схема работы одного из алгоритмов расчёта лимита ответственности на листе 2 графической части.

# Вычислительная система

## Требования к аппаратным и операционным ресурсам

Одним из главных вещей в разработке является компьютер. Написание проекта будет вестись на ноутбуке со следующими характеристиками:

* процессор Intel Core i7-1165G7, 2020 года выпуска, техпроцесс 10 нанометров, частота 2,8 ГГц и 4,7 ГГц в турбо режиме, 4 ядра, 8 потоков, 64х-битный набор команд;
* оперативная память 16 ГБ DDR4 2666 Гц;
* клавиатура и мышка.

## Инструменты разработки

При разработке программного средства будут использованы следующие программы:

* операционная система MS Windows 11 Home 64 разрядная;
* редактор кода MS Visual Studio;
* язык программирования C++;
* фреймворк MS Windows Forms.

MS Windows 11 была выбрана из-за необходимости тестирования на ней в будущем. Также выбор был обусловлен тем, что она более лучше подходит для мощных систем и является более приятной для использования, чем операционные системы семейства Linux.

MS Visual Studio 2022 был выбран из-за удобства пользования разработчика. Также данный редактор кода абсолютно бесплатный и не требует постоянной подписки. Также он является одним из лучших решений для разработки программ на языке программирования С++ на операционной системе MS Windows 11.

Для написания программы был выбран язык программирования C++. Это обусловлено его скоростью работы и целью закрепления изученного материала по данному языку программирования входе разработки программного средства.

MS Windows Forms – фреймворк программирования приложений, отвечающий за графический фреймворк пользователя и являющийся частью Microsoft .NET Framework. Данный фреймворк упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде.

# Проектирование задачи

## Требования к приложению

Основным способом защиты данных будет использование логинов пользователей. При создании пользователя в папке с программой будет создаваться 2 файла с расширением «txt», названием файла с логином будет выступать «логин», а названием файла с паролем будет выступать «пароль». Такая защита будет выстроена со стороны программы и пользователя, но в случае, если пользователю понадобится изменить что-либо без программы, то он может просто открыть и изменить необходимый ему файл.

Интерфейс программы должен быть интуитивно понятным для пользователей. Также у него должен быть выбор из 3 вариантов цвета фона окна, таких как:

* серый;
* синий;
* коралловый.

Пользователи также смогут использовать программу на трех языках, таких как:

* русский;
* белорусский;
* английский.

Для программы должны быть следующие аппаратные характеристики компьютера:

* процессор Intel Pentium P6200, 2012 года выпуска, техпроцесс 32 нанометра, частота 2,53 ГГц и 3,07 ГГц в турбо режиме, 2 ядра, 4 потока, 64-х битный набор команд;
* оперативная память 4 ГБ DDR4 1866 Гц;
* операционная система MS Windows 10, 64-х разрядная;

Доступ к интернету не будет необходим при работе в программе.

## Концептуальный прототип

Данный раздел направлен на описание и уточнение внешнего вида программы и демонстрации прототипа приложения.

На прототипе можно увидеть, что в левой части формы и верхнем меню программного средства будут находится такие пункты как:

* «Справка»;
* «Язык»;
* «Тема»;
* «ФИО страхователя»;
* «ФИО застрахованного»;
* «Пол»;
* «Дата рождения»;
* «Дата начала страхования»;
* «Периодичность уплаты взносов»;
* «Срок страхования (лет, месяцев)»;
* «Канал продаж»;
* «Тип страхователя»;
* «Возраст на дату начала (лет, месяцев)»;
* «Нагрузка»;
* «Норма доходности».

В свою очередь в правой части формы будут находится такие пункты как:

* «Закрыть»;
* «Достижение возраста»;
* «Смерть»;
* «Инвалидность»;
* «Вред здоровью»;
* «Временная нетрудоспособность»;
* «Страховой взнос»;
* «Лимит ответственности»;
* «Тариф»;
* «Нагрузка»;
* «Норма доходности»;
* «Налоговый вычет»;
* «Страховая сумма»;
* «Периодический взнос»;
* «Общий взнос по договору»;
* «Сохранение в txt-файл».

Все данные пункты будут отвечать за свои функции и распределены по форме следующим образом, представленным на рисунке 1.

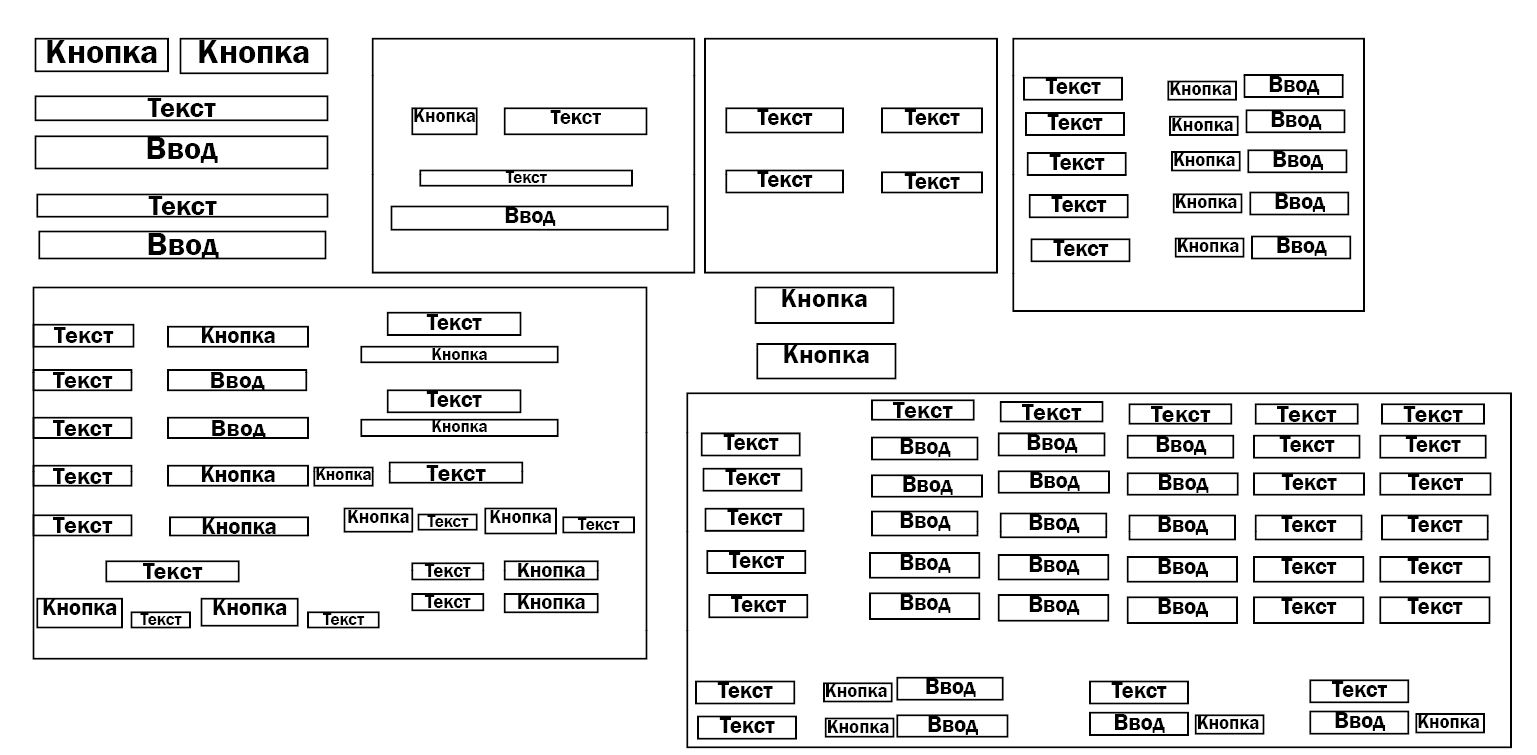


Рисунок 1 – Прототип главного окна программы

## Функции и элементы управления

Основываясь на вариантах использования, можно выделить некоторые основные функции программы, это такие как: расчет лимита ответственности исходя из введенных данных, смена темы программы, смена языка программы, выбор тарифа исходя из периодичности уплат.

Код расчета лимита ответственности исходя из введенных данных на примере страхового случая «достижение возраста».

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

double a = Convert::ToDouble(this->textBox42->Text);//конвертирование textBox42 double значения

double c = Convert::ToDouble(this->textBox26->Text); //конвертирование textBox26 double значения

this->textBox13->Clear();//очистить textBox13

this->textBox13->Text = a.ToString();//присвоение textBox13 значение из textBox42

this->Text = a.ToString();//отображение переданного значения

double b = a \* c;//расчет лимита ответственности по страховому случаю "достижение возраста"

this->textBox18->Clear();//очистить textBox18

this->textBox18->Text = b.ToString();//присвоение textBox18 значение b

this->Text = b.ToString();//отображение переданного значения

}

Смена темы программы служит для изменения ее внешнего вида и кастомизации под пользователя.

Код кнопки смены темы программы на серый цвет.

private: System::Void светлаяToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::LightSlateGray;//присвоение окну цвет «серый»

}

Смена темы программы служит для изменения ее внешнего вида и кастомизации под пользователя.

Код кнопки смены темы программы на синий цвет.

private: System::Void темнаяToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::DarkBlue;//присвоение окну цвет «синий»

}

Смена темы программы служит для изменения ее внешнего вида и кастомизации под пользователя.

Код кнопки смены темы программы на коралловый цвет.

private: System::Void креативнаяToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::LightCoral;//присвоение окну цвет «коралловый»

}

Смена языка программы служит для изменения локализации программы и кастомизации под пользователя.

Код кнопки смена языка программы на английский.

private: System::Void английскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Text = "The program for the selection of cumulative life insurance according to Rules No. 6";//присвоить главному окну текст

видToolStripMenuItem->Text = "View";//присвоить ToolStripMenu текст

темаToolStripMenuItem->Text = "Topic";// присвоить ToolStripMenu текст

языкToolStripMenuItem->Text = "Language";// присвоить ToolStripMenu текст

светлаяToolStripMenuItem->Text = "Grey";// присвоить ToolStripMenu текст

темнаяToolStripMenuItem->Text = "DarkBlue";// присвоить ToolStripMenu текст

креативнаяToolStripMenuItem->Text = "Coral";// присвоить ToolStripMenu текст

русскийToolStripMenuItem->Text = "Russian";// присвоить ToolStripMenu текст

английскийToolStripMenuItem->Text = "English";// присвоить ToolStripMenu текст

белорусскийToolStripMenuItem->Text = "Belarusian";// присвоить ToolStripMenu текст

помощьToolStripMenuItem->Text = "Help";// присвоить ToolStripMenu текст

описаниеToolStripMenuItem->Text = "Description";// присвоить ToolStripMenu текст

label1->Text = "Full name of the policyholder";//присвоение label1 текст

label2->Text = "Full name of the insured";// присвоение label2 текст

label3->Text = "The amount of the fee to be paid at the conclusion";// присвоение label3 текст

label6->Text = "Gender";// присвоение label6 текст

label7->Text = "Date of birth";// присвоение label7 текст

label8->Text = "Insurance start date";// присвоение label8 текст

label9->Text = "The frequency of payment of contributions";// присвоение label9 текст

label10->Text = "Insurance currency";// присвоение label10 текст

label11->Text = "The insurance period";// присвоение label11 текст

label15->Text = "Sales Channel";// присвоение label15 текст

label16->Text = "Type of policyholder";// присвоение label16 текст

label17->Text = "Age as of the start date";// присвоение label17 текст

label23->Text = "Load";// присвоение label23 текст

label24->Text = "The rate of return";// присвоение label24 текст

label27->Text = "Insurance";// присвоение label25 текст

label28->Text = "premium";// присвоение label28 текст

label29->Text = "Reaching the age";// присвоение label29 текст

label30->Text = "Death";// присвоение label30 текст

label31->Text = "Disability";// присвоение label31 текст

label32->Text = "Harm to health";// присвоение label32 текст

label33->Text = "Temporary disability";// присвоение label33 текст

label34->Text = "Limit";// присвоение label34 текст

label35->Text = "Periodic payment";// присвоение label35 текст

label36->Text = "The total contribution under the contract";// присвоение label36 текст

label37->Text = "responsibilities";// присвоение label37 текст

label38->Text = "Insurance amount";// присвоение label38 текст

label53->Text = "Tax deduction";// присвоение label53 текст

label54->Text = "Reaching the age"; присвоение label54 текст

label55->Text = "Death"; присвоение label55 текст

label56->Text = "Disability";// присвоение label56 текст

label57->Text = "Harm to health";// присвоение label57 текст

label58->Text = "Temporary disability";// присвоение label58 текст

groupBox1->Text = "\*Loyalty\* Program";//присвоение groupBox1 текст

groupBox2->Text = "Exchange rates";// присвоение groupBox2 текст

groupBox3->Text = "Information for the conclusion of the contract";// присвоение groupBox3 текст

groupBox4->Text = "Summary data";// присвоение groupBox4 текст

groupBox5->Text = "Information for calculation";// присвоение groupBox5 текст

button1->Text = "Save to txt file";//присвоить button1 текст

checkBox1->Text = "Apply the program";//присвоить checkBox1 текст

}

Код выбора тарифа исходя из периодичности уплат уплат.

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e)

{

String^ comboBoxValue = this->comboBox2->SelectedItem->ToString();

if (comboBoxValue == "Ежегодно")//если надпись в comboBox2 имеет название ежегодно, то присвоить textBox26/27/28/29/30 значение тарифа

{

double c = 18.1908; //присвоение переменной значения тарифа для страхового случая «Достижение возраста»

this->textBox26->Clear();//очистка textBox26

this->textBox26->Text = c.ToString();//присвоение значения переменной textBox26

double e = 0.0945;// присвоение переменной значения тарифа для страхового случая «Смерть»

this->textBox27->Clear();//очистка textBox27

this->textBox27->Text =e.ToString();//присвоение значения переменной textBox27

double f = 0.0045;//присвоение переменной значения тарифа для страхового случая «Инвалидность»

this->textBox28->Clear();//очистка textBox28

this->textBox28->Text = f.ToString();//присвоение значения переменной textBox28

double n = 2.0250;// присвоение переменной значения тарифа для страхового случая «Вред здоровью»

this->textBox29->Clear();//очистка textBox29

this->textBox29->Text = n.ToString();//присвоение значения переменной textBox29

double u = 2.4923; //присвоение переменной значения тарифа для страхового случая «Временная нетрудоспособность»

this->textBox30->Clear();//очистка textBox30

this->textBox30->Text = u.ToString();//присвоение значения переменной textBox30

}

else if(comboBoxValue == "Раз в полгода")// если надпись в comboBox2 имеет название раз в полгода, то присвоить textBox26/27/28/29/30 значение тарифа

{

double d = 9.2278;//присвоение переменной значения тарифа для страхового случая «Достижение возраста»

this->textBox26->Clear();//очистка textBox26

this->textBox26->Text = d.ToString();//присвоение значения переменной textBox26

double c = 0.0475;// присвоение переменной значения тарифа для страхового случая «Смерть»

this->textBox27->Clear();//очистка textBox27

this->textBox27->Text = c.ToString();//присвоение значения переменной textBox27

double y = 0.0025;// присвоение переменной значения тарифа для страхового случая «Инвалидность»

this->textBox28->Clear();//очистка textBox28

this->textBox28->Text = y.ToString();//присвоение значения переменной textBox28

double x = 1.0125;// присвоение переменной значения тарифа для страхового случая «Вред здоровью»

this->textBox29->Clear();//очистка textBox29

this->textBox29->Text = x.ToString();//присвоение значения переменной textBox29

double o = 1.2460;// присвоение переменной значения тарифа для страхового случая «Временная нетрудоспособность»

this->textBox30->Clear();//очистка textBox30

this->textBox30->Text = o.ToString();//присвоение значения переменной textBox30

}

else// если надпись в comboBox2 имеет название ежемесячно, то присвоить textBox26/27/28/29/30 значение тарифа

{

double j = 1.4713;// присвоение переменной значения тарифа для страхового случая «Достижение возраста»

this->textBox26->Clear();//очистка textBox26

this->textBox26->Text = j.ToString();//присвоение значения переменной textBox26

double s = 0.0079;// присвоение переменной значения тарифа для страхового случая «Смерть»

this->textBox27->Clear();//очистка textBox27

this->textBox27->Text = s.ToString();//присвоение значения переменной textBox27

double v = 0.0055;// присвоение переменной значения тарифа для страхового случая «Инвалидность»

this->textBox28->Clear();//очистка textBox28

this->textBox28->Text = v.ToString();//присвоение значения переменной textBox28

double x = 0.1688;// присвоение переменной значения тарифа для страхового случая «Вред здоровью»

this->textBox29->Clear();//очистка textBox29

this->textBox29->Text = x.ToString();//присвоение значения переменной textBox29

double z = 0.2080;// присвоение переменной значения тарифа для страхового случая «Временная нетрудоспособность»

this->textBox30->Clear();//очистка textBox30

this->textBox30->Text = z.ToString();//присвоение значения переменной textBox30

}

}

## Проектирование справочной системы приложения

Справочная система будет представлена в виде формы, которая будет открываться при нажатии на кнопку «Справка». Она была создана простым путем написания справки в текстовое поле в окне. Там будет идти текст о том, как необходимо пользоваться программой, также пользователь сможет выбрать язык, который ему больше всего подходит. (см. рисунок 2)

# Описание программного средства

## Общие сведения

Как уже говорилось в первом разделе программа будет вычислять сумму страховых взносов с учетом тарифов и периодичности уплат. Этот проект необходим для работников страховых компаний отвечающих за расчет страховых выплат и заключений договоров с клиентами.

Для запуска программы на данный момент потребуется операционная система MS Windows 10 64bit.

Также для запуска программы понадобится следующее минимальное аппаратное обеспечение необходимое для функционирования данного курсового проекта:

* процессор Intel Core i5 540m;
* оперативная память 4 ГБ DDR4;
* свободное место на диске 500 МБ;
* видеокарта AMD Radeon HD Graphics 7370m 2 ГБ.

Установщика программного средства не будет, в связи с тем, что программа сразу будет идти как исполняемый файл. Пользователь сможет скачать папку с программой и всеми необходимыми для нее файлами, уже находящиеся в папке.

## Входные и выходные данные

Входными данными являются файлы «txt». Из текстовых файлов программа будет получать информацию о логинах и паролей.

Выходные данные программы будут записываться в файлы «txt». Будут записываться данные, вводимые пользователем, данные для заключения договора с клиентом и расчета страховых взносов и общего взноса по договору с учетом налогового вычета.

# Методика испытаний

## Технические требования

Для программы должны быть следующие аппаратные характеристики компьютера:

* процессор Intel Core i5 540m, 2012 года выпуска, техпроцесс 32 нанометра, частота 2,53 ГГц и 3,07 ГГц в турбо режиме, 2 ядра, 4 потока, 64-х битный набор команд;
* оперативная память 4 ГБ DDR4 1066 Гц;
* операционная система MS Windows 10, 64-х разрядная.

Доступ к интернету не будет необходим при работе в программе.

Программных средств для функционирования программе не нужно.

## Функциональное тестирование

Тестирования является важным этапом при разработки программного средства. В тестировании есть несколько способов его проведения, но каждый этот способ объединяет наличие тест-кейсов при их проведении. Тест-кейсы для программы будут приведены в таблице 1.

Таблица 1 – Тест-кейсы

|  |  |  |  |
| --- | --- | --- | --- |
| Модуль / Функция | Шаги воспроизведения | Ожидаемый результат | Фактический результат |
| 1 | 2 | 3 | 4 |
| Регистрация | 1 Ввод логина  2 Ввод пароля  3 Нажать на кнопку «Сохранить» введенных данных | Данные сохраняются в txt-файл и далее пользователя перекидывает в окно для авторизации | После введения пользователем логина и пароля данные сохраняються в два отдельных файла “логин” и “пароль”, далее пользователя перекидывает на форму авторизации |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Авторизация | 1 Ввод логина  2 Ввод пароля  3 Нажать на кнопку «Войти» | Если пользователь был уже зарегистрирован, то с помощью авторизации пользователь может перейти на основную форму | После того, как пользователь ввел данные, то с помощью кнопки войти пользователь попадает на главную фому |
| Вычисление лимита ответственности | 1. 1 Ввести данные в поле со страховым случаем 2. 2 Нажать на кнопку, что бы вычислить | В окно «Результат» записывается рассчитанный лимит ответственности по страховому случаю | Ответ вычислений лимита ответственности исходя из введенных данных, умноженных на тариф. |
| Расчет общего взноса по договору | 1 Выбрать на сколько лет страховка  2 Выбор на сколько месяцев заключается договор (не обязательно)  3 Нажать на кнопку чтобы рассчитать | В окно «Результат» записывается ответ расчета общего взноса по договору | Ответ рассчитан с помощью умножения выбранного срока заключения договора на периодический взнос |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| Расчет тарифа | 1Выбрать подходящую клиенту периодичность уплат  2 Нажать на кнопку, для расчета нужного тарифа | В окно выводится фиксированный тариф, подходящий выбранной периодичности уплат | Ответ выводится в окно исходя из выбранной периодичности уплат |
| Расчет периодического взноса | 1 Нажать на кнопку для расчета периодического взноса | В окно выводится результат подсчета периодического взноса | Ответ выводится в окно исходя из суммирования всех страховых взносов по данным страховым случаям |
| Сохранение введенных данных в файл txt | 1 Нажать на кнопку «Сохранение в txt-файл» | Данные сохраняются в формате .txt в папку курсач\_2курс | Данные сохраняются в формате .txt в папку курсач\_2курс |
| Расчет страховой суммы | 1 Нажать на кнопку для расчета страховой суммы | В окно выводится результат подсчета страховой суммы | Ответ выводится в окно исходя из суммирования лимитов ответственности по данным страховым случаям |
| Справка | 1 Нажать на главной форме кнопку «Справка» | В форме будет отображается текст справки | После нажатия на кнопку на главной форме появляется форма «Справка» в которой отображается текст данной справки |

# Применение

## Назначение программы

Программное средство имеет следующее название «Программное средство для расчета страховых накопительных выплат». Она может использоваться для расчета накопительных выплат, а также упрощения расчета периодических взносов. Данное программное средство, как уже говорилось, должно автоматизировать и упростить процесс подсчета накопительных страховых взносов исходя из тарифов, периодичности уплат и взносов по страховым случаям. К сожалению, страховую выплату можно рассчитать только по окончанию срока страхования, поскольку в течении срока страхования страхователь может изменить размер страхового взноса, как в сторону уменьшения, так и увеличения. Поэтому, исходя из данных факторов, я в своем программном средстве произвела расчет только страховых взносов и периодичность уплаты взносов.

Сетевая поддержка в программе не требуется, так как программа полностью локальная и не использует сервер.

## Условия применения

Для корректной работы программы нет необходимости в сторонних программных продуктах или средствах. Программа является полностью автономной и может переноситься без особого труда на разные компьютеры.

Также для работы программы у компьютера должны следующие минимальные аппаратные характеристики:

* процессор Intel Core i5 540m, 2012 года выпуска, техпроцесс 32 нанометра, частота 2,53 ГГц и 3,07 ГГц в турбо режиме, 2 ядра, 4 потока, 64х- битный набор команд;
* оперативная память 4 ГБ DDR4 1066 Гц;
* операционная система MS Windows 10, 64х-разрядная;
* клавиатура;
* мышь.

Данные аппаратные характеристики основаны на примере компьютера, на котором велась разработка программного средства.

## Справочная система

Справочная система будет представлена в виде формы, которая будет открываться при нажатии на кнопку «Справка». Она была создана простым путем написания справки в текстовое поле в окне. Там будет идти текст о том, как необходимо пользоваться программой. Форма справочной информации предоставлена на рисунке 2.

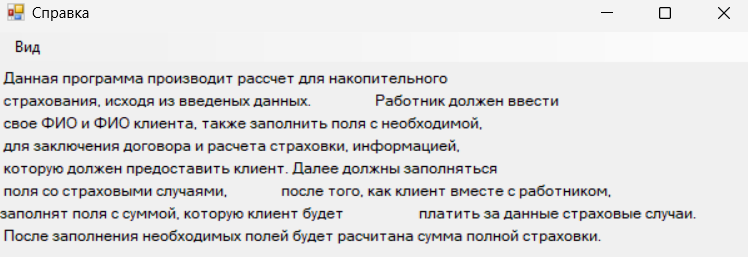


Рисунок 2 – Форма справочной информации

# Заключение

По итогам создания курсового проекта на тему «Программное средство для расчета страховых накопительных выплат» было разработано программное средство «курсовая\_2курс.exe».

Основной и главной функцией программы является расчет накопительных выплат.

Для достижения целей курсового проекта были решены следующие задачи:

* изучена предметная область;
* разработана физическая и логическая модель данных;
* разработано программное средство;
* описана область применения, создание программного средства.

Интерфейс программы является минималистичным и интуитивно понятным.

В ходе реализации данного программного продукта сложностей, кроме нахождения нужных для программного средства формул, не возникало. В программе реализовано большинство требования и продукт является полностью отлаженным.

К особенностям программы можно отнести простоту интерфейса, удобство работы и малые затраты на обучение работы с программным средством.

В процессе разработки программного продукта были закреплены и получены знания по уже известному материалу, отработаны навыки владения методами устойчивого программирования и эффективной разработки в MS Visual Studio 2022, с использованием языка программирования C++, Фреймворка MS Windows Form. Также использование файла с расширением «txt» предназначенного для хранения введенных данных.

# Список информационных источников

1. Дейтел, Х. Как программировать на C++ / Х. Дейтел, П. Дейтел. – М.: ДМК Пресс, 2018. – 1021 с.
2. Мейерс, С. Эффективное использование С++. 55 верных способов улучшить структуру и код ваших программ / С. Мейерс. – М.: ДМК Пресс, 2006. – 300 с.
3. Общие требования к тестовым документам : ГОСТ 2.105-95. – Введ. 01.01.1996. – Минск : Межгос. совет по стандартизации, метрологии и сертификации, 1995. – 84 с.
4. Объектно-ориентированный анализ и проектирование с примерами приложений / Гради Буч [и др.]. – 3-е изд. – М. : ООО «И.Д. Вильямс», 2008. – 720 с.
5. Описание программы. Требования к содержанию, оформлению и контролю качества: ГОСТ 19.402-2000. – Введ. 01.09.2001. – Минск: Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 14 с.
6. Программа и методика испытаний. Требования к содержанию, оформлению и контролю качества: ГОСТ 19.301-2000. – Введ. 01.09.2001. – Минск: Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 14 с.
7. Руководство по программированию на C+ [Электронный ресурс]. – Microsoft, 2019. – Режим доступа : http://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/. – Дата доступа : 25.04.2023.
8. Текст программы. Требования к содержанию, оформлению и контролю качества: ГОСТ 19.401-2000. – Введ. 01.09.2001. – Минск: Межгос. совет по стандартизации, метрологии и сертификации, 2000. – 16 с.
9. Тепляков, С. Паттерны проектирования на платформе .NET / С.Тепляков. – СПб. : Питер, 2015. – 320 с.
10. Шаляпин, Ю.В. Основы алгоритмизации и программирование: методические указания к курсовому проектированию для учащихся специальности 2-40 01 01 «Программное обеспечение информационных технологий» / Ю.В. Шаляпин. – Минск: ЧУО «КБП», 2022. – 19 с.

# ПРИЛОЖЕНИЕ А

**(обязательное)**

**Текст программы**

//MyForm.h

#pragma once

#include "MyForm2.h"

#include "MyForm1.h"

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

using namespace System::IO;

namespace курсовая2курс {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

public ref class MyForm : public System::Windows::Forms::Form

{

public:

MyForm(void)

{

InitializeComponent();

this->StartPosition = FormStartPosition::CenterScreen;

}

protected:

~MyForm()

{

if (components)

{

delete components;

}

}

#pragma endregion

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Close();

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

MyForm1^ form = gcnew MyForm1();

form->Show();

this->Hide();

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e) {

String^ value = textBox1->Text;

String^ value2 = textBox2->Text;

// Создаем экземпляр StreamWriter для записи в файл

StreamWriter^ writer = gcnew StreamWriter("пароль.txt");

StreamWriter^ writer2 = gcnew StreamWriter("логин.txt");

// Записываем значение в файл

writer->WriteLine(value);

writer2->WriteLine(value2);

// Закрываем StreamWriter

writer->Close();

writer2->Close();

// Выводим сообщение об успешном сохранении

MessageBox::Show("Данные сохранены");

MyForm1^ form = gcnew MyForm1();

form->Show();

this->Hide();

}

private: System::Void розовыйToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::LightPink;//сменить цевт окна

button1->BackColor = Color::Aqua;//сменить цвет кнопки

button2->BackColor = Color::Aqua;//сменить цвет кнопки

button3->BackColor = Color::Aqua;// сменить цвет кнопки

}

private: System::Void синийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::PeachPuff;//сменить цвет окна

button1->BackColor = Color::LightSkyBlue;// сменить цвет кнопки

button2->BackColor = Color::LightSkyBlue;// сменить цвет кнопки

button3->BackColor = Color::LightSkyBlue;// сменить цвет кнопки

}

private: System::Void фиолетовыйToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::MediumPurple;// сменить цвет окна

button1->BackColor = Color::LimeGreen;//сменить цвет кнопки

button2->BackColor = Color::LimeGreen;// сменить цвет кнопки

button3->BackColor = Color::LimeGreen;// сменить цвет кнопки

}

private: System::Void русскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

label1->Text = "РЕГИСТРАЦИЯ"; //смена языка текста на русский

label2->Text = "ЛОГИН";//смена языка текста на русский

label3->Text = "ПАРОЛЬ";//смена языка текста на русский

button1->Text = "ВЫХОД";// смена языка кнопки на русский

button2->Text = "АВТОРИЗАЦИЯ";// смена языка кнопки на русский

button3->Text = "ОК";// смена языка кнопки нарусский

видToolStripMenuItem->Text = "Вид";

темаToolStripMenuItem->Text = "Тема";

языкToolStripMenuItem->Text = "Язык";

розовыйToolStripMenuItem->Text = "Розовый";

синийToolStripMenuItem->Text = "Персиковый";

фиолетовыйToolStripMenuItem->Text = "Фиолетовый";

русскийToolStripMenuItem->Text = "Русский";

английскийToolStripMenuItem->Text = "Английский";

белорусскийToolStripMenuItem->Text = "Белорусский";

}

private: System::Void видToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void английскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

label1->Text = "REGISTRATION";

label2->Text = "LOGIN";

label3->Text = "PASSWORD";

button1->Text = "EXIT";

button2->Text = "AUTHORIZATION";

button3->Text = "OK";

видToolStripMenuItem->Text = "VIEW";

темаToolStripMenuItem->Text = "Topic";

языкToolStripMenuItem->Text = "Language";

розовыйToolStripMenuItem->Text = "Pink";

синийToolStripMenuItem->Text = "Peach";

фиолетовыйToolStripMenuItem->Text = "Purpule";

русскийToolStripMenuItem->Text = "Russsian";

английскийToolStripMenuItem->Text = "English";

белорусскийToolStripMenuItem->Text = "Belarusian";

}

private: System::Void белорусскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

label1->Text = "РЭГІСТРАЦЫЯ";

label2->Text = "ЛОГІН";

label3->Text = "ПАРОЛЬ";

button1->Text = "ВЫХАД";

button2->Text = "АЎТАРЫЗАЦЫЯ";

button3->Text = "ОК";

видToolStripMenuItem->Text = "Выгляд";

темаToolStripMenuItem->Text = "Тэма";

языкToolStripMenuItem->Text = "Мова";

розовыйToolStripMenuItem->Text = "Ружовы";

синийToolStripMenuItem->Text = "Персікавы";

фиолетовыйToolStripMenuItem->Text = "Фіялетавы";

русскийToolStripMenuItem->Text = "Рускі";

английскийToolStripMenuItem->Text = "Англійскі";

белорусскийToolStripMenuItem->Text = "Беларускі";

}

};

}

**//** MyForm1.h

#pragma once

#include "MyForm.h"

#include "MyForm2.h"

#include <iostream>

#include <fstream>

#include <string>

#include <Windows.h>

#include <msclr/marshal\_cppstd.h>

using namespace std;

using namespace System::IO;

using namespace msclr::interop;

namespace курсовая2курс {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

public ref class MyForm1 : public System::Windows::Forms::Form

{

public:

MyForm1(void)

{

InitializeComponent();

}

protected:

~MyForm1()

{

if (components)

{

delete components;

}

}

#pragma endregion

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Close();

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

StreamReader^ reader = gcnew StreamReader("логин.txt");

StreamReader^ reader2 = gcnew StreamReader("пароль.txt");

// Читаем первую строку из файла

String^ savedValue = reader->ReadLine();

String^ savedValue2 = reader2->ReadLine();

// Закрываем StreamReader

reader->Close();

reader2->Close();

// Получаем значение из текстового поля

String^ currentValue = textBox1->Text;

String^ currentValue2 = textBox2->Text;

// Сравниваем значения

if (savedValue == currentValue && savedValue2 == currentValue2)

{

// Выводим сообщение об успешной проверке

MyForm2^ form1 = gcnew MyForm2();

form1->Show();

this->Hide();

}

else

{

// Выводим сообщение об ошибке

MessageBox::Show("Данные не найдены");

}

}

private: System::Void красныйToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::Red;

button1->BackColor = Color::HotPink;

button2->BackColor = Color::HotPink;

}

private: System::Void желтыйToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::Orange;

button1->BackColor = Color::IndianRed;

button2->BackColor = Color::IndianRed;

}

private: System::Void зеленыйToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::LightGreen;

button1->BackColor = Color::Olive;

button2->BackColor = Color::Olive;

}

private: System::Void русскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

label1->Text = "АВТОРИЗАЦИЯ";

label2->Text = "ЛОГИН";

label3->Text = "ПАРОЛЬ";

button1->Text = "ВЫХОД";

button2->Text = "ОК";

видToolStripMenuItem->Text = "Вид";

темаToolStripMenuItem->Text = "Тема";

языкToolStripMenuItem->Text = "Язык";

красныйToolStripMenuItem->Text = "Красный";

желтыйToolStripMenuItem->Text = "Желтый";

зеленыйToolStripMenuItem->Text = "Зеленый";

русскийToolStripMenuItem->Text = "Русский";

английскийToolStripMenuItem->Text = "Английский";

белорусскийToolStripMenuItem->Text = "Белорусский";

}

private: System::Void английскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

label1->Text = "AUTHORIZATION"; //Перевод label на английский

label2->Text = "LOGIN";// Перевод label на английский

label3->Text = "PASSWORD";// Перевод label на английский

button1->Text = "EXIT";// Перевод button на английский

button2->Text = "OK";// Перевод button на английский

видToolStripMenuItem->Text = "VIEW";// Перевод ToolStripMenu на английский

темаToolStripMenuItem->Text = "Topic";// Перевод ToolStripMenu на английский

языкToolStripMenuItem->Text = "Language";// Перевод ToolStripMenu на английский

красныйToolStripMenuItem->Text = "Red";// Перевод ToolStripMenu на английский

желтыйToolStripMenuItem->Text = "Orange";// Перевод ToolStripMenu на английский

зеленыйToolStripMenuItem->Text = "Green";// Перевод ToolStripMenu на английский

русскийToolStripMenuItem->Text = "Russsian";// Перевод ToolStripMenu на английский

английскийToolStripMenuItem->Text = "English";// Перевод ToolStripMenu на английский

белорусскийToolStripMenuItem->Text = "Belarusian";// Перевод ToolStripMenu на английский

}

private: System::Void белорусскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

label1->Text = "АЎТАРЫЗАЦЫЯ";//Перевод label на белорусский

label2->Text = "ЛОГІН";// Перевод label на белорусский

label3->Text = "ПАРОЛЬ";// Перевод label на белорусский

button1->Text = "ВЫХАД";// Перевод button на белорусский

button2->Text = "ОК";// Перевод button на белорусский

видToolStripMenuItem->Text = "Выгляд";// Перевод ToolStripMenu на белорусский

темаToolStripMenuItem->Text = "Тэма";// Перевод ToolStripMenu на белорусский

языкToolStripMenuItem->Text = "Мова";// Перевод ToolStripMenu на белорусский

красныйToolStripMenuItem->Text = "Чырвоны";// Перевод ToolStripMenu на белорусский

желтыйToolStripMenuItem->Text = "Жоўты";// Перевод ToolStripMenu на белорусский

зеленыйToolStripMenuItem->Text = "Зялены";// Перевод ToolStripMenu на белорусский

русскийToolStripMenuItem->Text = "Рускі";// Перевод ToolStripMenu на белорусский

английскийToolStripMenuItem->Text = "Англійскі";// Перевод ToolStripMenu на белорусский

белорусскийToolStripMenuItem->Text = "Беларускі";// Перевод ToolStripMenu на белорусский

}

private: System::Void MyForm1\_Load(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void textBox2\_TextChanged(System::Object^ sender, System::EventArgs^ e)

{

}

};

}

//MyForm2.h

#pragma once

#include "MyForm3.h"//библиотека формы

#include <iostream>

#include <fstream> //библиотека для работы с файлами

#include <string>//библиотека для работы со строками

#include <Windows.h>

#include <msclr/marshal\_cppstd.h> //библиотека для конвертации строк

using namespace std;

using namespace System::IO;

using namespace msclr::interop;

namespace курсовая2курс {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

public ref class MyForm2 : public System::Windows::Forms::Form

{

public:

MyForm2(void)

{

InitializeComponent();

}

protected:

~MyForm2()

{

if (components)

{

delete components;

}

}

#pragma endregion

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Close();

}

private: System::Void светлаяToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::LightSlateGray;

}

private: System::Void темнаяToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::DarkBlue;

}

private: System::Void креативнаяToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->BackColor = Color::LightCoral;

}

private: System::Void русскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Text = "Программа по подбору накопительного страхования жизни по Правилам №6";

видToolStripMenuItem->Text = "Вид";

темаToolStripMenuItem->Text = "Тема";

языкToolStripMenuItem->Text = "Язык";

светлаяToolStripMenuItem->Text = "Серый";

темнаяToolStripMenuItem->Text = "Синий";

креативнаяToolStripMenuItem->Text = "Кораловый";

русскийToolStripMenuItem->Text = "Русский";

английскийToolStripMenuItem->Text = "Английский";

белорусскийToolStripMenuItem->Text = "Белорусский";

помощьToolStripMenuItem->Text = "Помощь";

описаниеToolStripMenuItem->Text = "Описание";

label1->Text = "ФИО страхователя";

label2->Text = "ФИО застрахованного";

label3->Text = "Сумма уплачиваемого взноса при заключении";

label4->Text = "Доллар США";

label5->Text = "ЕВРО";

label6->Text = "Пол";

label7->Text = "Дата рождения";

label8->Text = "Дата начала стрхования";

label9->Text = "Периодичность уплаты взносов";

label10->Text = "Валюта страхования";

label11->Text = "Срок страхования";

label15->Text = "Канал продаж";

label16->Text = "Тип страхователя";

label17->Text = "Возраст на дату начала";

label23->Text = "Нагрузка";

label24->Text = "Норма доходности";

label27->Text = "Страховой";

label28->Text = "взнос";

label29->Text = "Достижение возраста";

label30->Text = "Смерть";

label31->Text = "Инвалидность";

label32->Text = "Вред здоровью";

label33->Text = "Временная нетрудоспособность";

label34->Text = "Лимит";

label35->Text = "Периодический взнос";

label36->Text = "Общий взнос по договору";

label37->Text = "ответственности";

label38->Text = "Страховая сумма";

label39->Text = "Тариф";

label40->Text = "Нагрузка";

label46->Text = "Норма";

label47->Text = "доходности";

label53->Text = "Налоговый вычет";

label54->Text = "Достижение возраста";

label55->Text = "Смерть";

label56->Text = "Инвалидность";

label57->Text = "Вред здоровью";

label58->Text = "Временная нетрудоспособность";

groupBox1->Text = "Программа \*Лояльность\*";

groupBox2->Text = "Курсы валют";

groupBox3->Text = "Информация для заключения договора";

groupBox4->Text = "Итоговые данные";

groupBox5->Text = "Информация для расчета";

button1->Text = "Созранить в txt-файл";

button2->Text = "Закрыть";

checkBox1->Text = "Применить программу";

}

private: System::Void MyForm2\_Load(System::Object^ sender, System::EventArgs^ e)

{заполнение comboBox1-9

//начало comboBox с 0

comboBox1->SelectedIndex = 0;

comboBox2->SelectedIndex = 0;

comboBox3->SelectedIndex = 0;

comboBox4->SelectedIndex = 0;

comboBox5->SelectedIndex = 0;

comboBox6->SelectedIndex = 0;

comboBox7->SelectedIndex = 0;

comboBox8->SelectedIndex = 0;

comboBox9->SelectedIndex = 0;

// очистить список элементов comboBox1

comboBox1->Items->Clear();

comboBox2->Items->Clear();

comboBox3->Items->Clear();

comboBox4->Items->Clear();

comboBox5->Items->Clear();

comboBox6->Items->Clear();

comboBox7->Items->Clear();

comboBox8->Items->Clear();

comboBox9->Items->Clear();

// сформировать список элементов comboBox1

comboBox1->Items->Add("Женский");

comboBox1->Items->Add("Мужской");

//сформировать список элементов comboBox2

comboBox2->Items->Add("Ежегодно");

comboBox2->Items->Add("Раз в полгода");

comboBox2->Items->Add("Ежемесячно");

//сформировать список элементов comboBox3

comboBox3->Items->Add("BYR");

comboBox3->Items->Add("USD");

comboBox3->Items->Add("EUR");

//сформировать список элементов comboBox4

comboBox4->Items->Add("3");

comboBox4->Items->Add("4");

comboBox4->Items->Add("5");

comboBox4->Items->Add("6");

comboBox4->Items->Add("7");

comboBox4->Items->Add("8");

comboBox4->Items->Add("9");

comboBox4->Items->Add("10");

//сформировать список элементов comboBox5

comboBox5->Items->Add("1");

comboBox5->Items->Add("2");

comboBox5->Items->Add("3");

comboBox5->Items->Add("4");

comboBox5->Items->Add("5");

comboBox5->Items->Add("6");

comboBox5->Items->Add("7");

comboBox5->Items->Add("8");

comboBox5->Items->Add("9");

comboBox5->Items->Add("10");

comboBox5->Items->Add("11");

//сформировать список элементов comboBox6

comboBox6->Items->Add("Прямые продажи");

//сформировать список элементов comboBox7

comboBox7->Items->Add("Физическое лицо");

//сформировать список элементов comboBox8

comboBox8->Items->Add("8,00");

//сформировать список элементов comboBox9

comboBox9->Items->Add("6,00");

}

private: System::Void groupBox3\_Enter(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void белорусскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Text = "Праграма па падобы назапашвальнага страхавання жыцця па правілах №6";

видToolStripMenuItem->Text = "Выгляд"; //Перевод ToolStripMenu на белорусский

темаToolStripMenuItem->Text = "Тэма";// Перевод ToolStripMenu на белорусский

языкToolStripMenuItem->Text = "Мова";// Перевод ToolStripMenu на белорусский

светлаяToolStripMenuItem->Text = "Шэры";// Перевод ToolStripMenu на белорусский

темнаяToolStripMenuItem->Text = "Сіні";// Перевод ToolStripMenu на белорусский

креативнаяToolStripMenuItem->Text = "Каралавы";// Перевод ToolStripMenu на белорусский

русскийToolStripMenuItem->Text = "Рускій";// Перевод ToolStripMenu на белорусский

английскийToolStripMenuItem->Text = "Англійскій";// Перевод ToolStripMenu на белорусский

белорусскийToolStripMenuItem->Text = "Беларускій";// Перевод ToolStripMenu на белорусский

помощьToolStripMenuItem->Text = "Дапамога";// Перевод ToolStripMenu на белорусский

описаниеToolStripMenuItem->Text = "Апісанне";// Перевод ToolStripMenu на белорусский

label1->Text = "Прозвішча, імя страхавальніка";// Перевод label на белорусский

label2->Text = "Прозвішча, імя застрахаванай";// Перевод label на белорусский

label3->Text = "Сумма выплачанага ўзносу пры заключэнні";// Перевод label на белорусский

label4->Text = "Долар ЗША";// Перевод label на белорусский

label5->Text = "ЕВРО";// Перевод label на белорусский

label6->Text = "Пол";// Перевод label на белорусский

label7->Text = "Дата нараджэння";// Перевод label на белорусский

label8->Text = "Дата пачатку стрхавання";// Перевод label на белорусский

label9->Text = "Перыядычнасць выплат ўзносаў";// Перевод label на белорусский

label10->Text = "Валюта страховання";// Перевод label на белорусский

label11->Text = "Тэрмін страховання";// Перевод label на белорусский

label15->Text = "Канал продажаў";// Перевод label на белорусский

label16->Text = "Тып страхавальніка";// Перевод label на белорусский

label17->Text = "Узрост на дату начала";// Перевод label на белорусский

label23->Text = "Нагрузка";// Перевод label на белорусский

label24->Text = "Норма даходнасці";// Перевод label на белорусский

label27->Text = "Страхавы";// Перевод label на белорусский

label28->Text = "ўзнос";// Перевод label на белорусский

label29->Text = "Дасягненне ўзросту";// Перевод label на белорусский

label30->Text = "Смерць";// Перевод label на белорусский

label31->Text = "Інваліднасть";// Перевод label на белорусский

label32->Text = "Шкода здароўю";// Перевод label на белорусский

label33->Text = "Часовая непрацаздольнасць";// Перевод label на белорусский

label34->Text = "Ліміт";// Перевод label на белорусский

label35->Text = "Перыядычны ўзнос";// Перевод label на белорусский

label36->Text = "Агульны ўзнос па дагавору";// Перевод label на белорусский

label37->Text = "адказнасці";// Перевод label на белорусский

label38->Text = "Страхавая сума";// Перевод label на белорусский

label47->Text = "даходнасці";// Перевод label на белорусский

label53->Text = "Падатковы вылік";// Перевод label на белорусский

label54->Text = "Дасягненне ўзросту";// Перевод label на белорусский

label55->Text = "Смерць";// Перевод label на белорусский

label56->Text = "Інваліднасць";// Перевод label на белорусский

label57->Text = "Шкода здароўю";// Перевод label на белорусский

label58->Text = "Часовая непрацаздольнасць";// Перевод label на белорусский

groupBox1->Text = "Праграма \*Лаяльнасць\*";// Перевод groupBox на белорусский

groupBox2->Text = "Курсы валют";// Перевод groupBox на белорусский

groupBox3->Text = "Інфармацыя для заключэння дагавора";// Перевод groupBox на белорусский

groupBox4->Text = "Выніковыя дадзеныя";// Перевод groupBox на белорусский

groupBox5->Text = "Інфармацыя для разліку";// Перевод groupBox на белорусский

button1->Text = "Захаваць ў txt-файл";// Перевод button на белорусский

button2->Text = "Зачыніць";// Перевод button на белорусский

checkBox1->Text = "Прымяніць праграму";// Перевод checkBox на белорусский

}

private: System::Void английскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Text = "The program for the selection of cumulative life insurance according to Rules No. 6";

видToolStripMenuItem->Text = "View";

темаToolStripMenuItem->Text = "Topic";

языкToolStripMenuItem->Text = "Language";

светлаяToolStripMenuItem->Text = "Grey";

темнаяToolStripMenuItem->Text = "DarkBlue";

креативнаяToolStripMenuItem->Text = "Coral";

русскийToolStripMenuItem->Text = "Russian";

английскийToolStripMenuItem->Text = "English";

белорусскийToolStripMenuItem->Text = "Belarusian";

помощьToolStripMenuItem->Text = "Help";

описаниеToolStripMenuItem->Text = "Description";

label1->Text = "Full name of the policyholder";

label2->Text = "Full name of the insured";

label3->Text = "The amount of the fee to be paid at the conclusion";

label4->Text = "USD";

label5->Text = "EUR";

label6->Text = "Gender";

label7->Text = "Date of birth";

label8->Text = "Insurance start date";

label9->Text = "The frequency of payment of contributions";

label10->Text = "Insurance currency";

label15->Text = "Sales Channel";

label16->Text = "Type of policyholder";

label17->Text = "Age as of the start date";

label23->Text = "Load";

label24->Text = "The rate of return";

label27->Text = "Insurance";

label28->Text = "premium";

label29->Text = "Reaching the age";

label30->Text = "Death";

label31->Text = "Disability";

label32->Text = "Harm to health";

label33->Text = "Temporary disability";

label34->Text = "Limit";

label35->Text = "Periodic payment";

label36->Text = "The total contribution under the contract";

label37->Text = "responsibilities";

label38->Text = "Insurance amount";

label53->Text = "Tax deduction";

label54->Text = "Reaching the age";

label55->Text = "Death";

label56->Text = "Disability";

label57->Text = "Harm to health";

label58->Text = "Temporary disability";

groupBox1->Text = "\*Loyalty\* Program";

groupBox2->Text = "Exchange rates";

groupBox3->Text = "Information for the conclusion of the contract";

groupBox4->Text = "Summary data";

groupBox5->Text = "Information for calculation";

button1->Text = "Save to txt file";

button2->Text = "Close";

checkBox1->Text = "Apply the program";

}

private: System::Void comboBox1\_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

String^ a = label1->Text;

String^ b = textBox1->Text;

String^ c = label2->Text;

String^ d = textBox2->Text;

String^ f = label6->Text;

String^ g = comboBox1->Text;

String^ h = label7->Text;

String^ i = textBox6->Text;

String^ j = textBox7->Text;

String^ k = label8->Text;

String^ l = label9->Text;

String^ y = comboBox6->Text;

String^ z = label16->Text;

String^ ez = comboBox7->Text;

String^ ab = label17->Text;

String^ bc = textBox9->Text;

String^ cd = label18->Text;

String^ de = textBox10->Text;

String^ ef = label19->Text;

String^ kl = label23->Text;

String^ lm = comboBox8->Text;

String^ pupa = label36->Text;

String^ bcb = textBox24->Text;

StreamWriter^ Write = gcnew StreamWriter("Inf.txt");

Write->WriteLine(a);

Write->WriteLine(b);

Write->WriteLine(c);

Write->WriteLine(d);

Write->WriteLine(ez);

Write->WriteLine(ab);

Write->WriteLine(bc);

Write->WriteLine(cd);

Write->WriteLine(de);

Write->WriteLine(zz);

Write->WriteLine(aa);

Write->WriteLine(bb);

Write->WriteLine(pipa);

Write->WriteLine(bib);

Write->WriteLine(ppa);

Write->WriteLine(bub);

Write->WriteLine(pupa);

Write->WriteLine(bcb);

Write->Close();

MessageBox::Show(this, "Даннные сохранены", "Сообщение", MessageBoxButtons::OK, MessageBoxIcon::Information);

}

private: System::Void описаниеToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e)

{

MyForm3^ form3 = gcnew MyForm3();

form3->Owner = this;

form3->Show();

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e)

{//расчет тарифа исходя из выбраной переодичности уплаты взносов

String^ comboBoxValue = this->comboBox2->SelectedItem->ToString();

if (comboBoxValue == "Ежегодно")//если надпись в comboBox2 имеет название ежегодно, то присвоить textBox26/27/28/29/30 значение тарифа

{

double c = 18.1908; //присвоение переменной значения тарифа для страхового случая «Достижение возраста»

this->textBox26->Clear();//очистка textBox26

this->textBox26->Text = c.ToString();//присвоение значения переменной textBox26

double e = 0.0945;// присвоение переменной значения тарифа для страхового случая «Смерть»

this->textBox27->Clear();//очистка textBox27

this->textBox27->Text =e.ToString();//присвоение значения переменной textBox27

double f = 0.0045;//присвоение переменной значения тарифа для страхового случая «Инвалидность»

this->textBox28->Clear();//очистка textBox28

this->textBox28->Text = f.ToString();//присвоение значения переменной textBox28

double n = 2.0250;// присвоение переменной значения тарифа для страхового случая «Вред здоровью»

this->textBox29->Clear();//очистка textBox29

this->textBox29->Text = n.ToString();//присвоение значения переменной textBox29

double u = 2.4923; //присвоение переменной значения тарифа для страхового случая «Временная нетрудоспособность»

this->textBox30->Clear();//очистка textBox30

this->textBox30->Text = u.ToString();//присвоение значения переменной textBox30

}

else if(comboBoxValue == "Раз в полгода")// если надпись в comboBox2 имеет название раз в полгода, то присвоить textBox26/27/28/29/30 значение тарифа

{

double d = 9.2278;//присвоение переменной значения тарифа для страхового случая «Достижение возраста»

this->textBox26->Clear();//очистка textBox26

this->textBox26->Text = d.ToString();//присвоение значения переменной textBox26

double c = 0.0475;// присвоение переменной значения тарифа для страхового случая «Смерть»

this->textBox27->Clear();//очистка textBox27

this->textBox27->Text = c.ToString();//присвоение значения переменной textBox27

double y = 0.0025;// присвоение переменной значения тарифа для страхового случая «Инвалидность»

this->textBox28->Clear();//очистка textBox28

this->textBox28->Text = y.ToString();//присвоение значения переменной textBox28

double x = 1.0125;// присвоение переменной значения тарифа для страхового случая «Вред здоровью»

this->textBox29->Clear();//очистка textBox29

this->textBox29->Text = x.ToString();//присвоение значения переменной textBox29

double o = 1.2460;// присвоение переменной значения тарифа для страхового случая «Временная нетрудоспособность»

this->textBox30->Clear();//очистка textBox30

this->textBox30->Text = o.ToString();//присвоение значения переменной textBox30

}

else// если надпись в comboBox2 имеет название ежемесячно, то присвоить textBox26/27/28/29/30 значение тарифа

{

double j = 1.4713;// присвоение переменной значения тарифа для страхового случая «Достижение возраста»

this->textBox26->Clear();//очистка textBox26

this->textBox26->Text = j.ToString();//присвоение значения переменной textBox26

double s = 0.0079;// присвоение переменной значения тарифа для страхового случая «Смерть»

this->textBox27->Clear();//очистка textBox27

this->textBox27->Text = s.ToString();//присвоение значения переменной textBox27

double v = 0.0055;// присвоение переменной значения тарифа для страхового случая «Инвалидность»

this->textBox28->Clear();//очистка textBox28

this->textBox28->Text = v.ToString();//присвоение значения переменной textBox28

double x = 0.1688;// присвоение переменной значения тарифа для страхового случая «Вред здоровью»

this->textBox29->Clear();//очистка textBox29

this->textBox29->Text = x.ToString();//присвоение значения переменной textBox29

double z = 0.2080;// присвоение переменной значения тарифа для страхового случая «Временная нетрудоспособность»

this->textBox30->Clear();//очистка textBox30

this->textBox30->Text = z.ToString();//присвоение значения переменной textBox30

}

}

private: System::Void checkBox1\_CheckedChanged(System::Object^ sender, System::EventArgs^ e) {

if (checkBox1->Checked)

{

MessageBox::Show(this,"К сожалению на данный момент акция отсутствует! Приносим свои извенения!", "Сообщение",MessageBoxButtons::OK, MessageBoxIcon::Information);

}

}

private: System::Void groupBox5\_Enter(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e) {

double a = Convert::ToDouble(this->textBox42->Text);

double c = Convert::ToDouble(this->textBox26->Text);

this->textBox13->Clear();

this->textBox13->Text = a.ToString();

double b = a \* c;//расчет лимита ответственности по страховому случаю "достижение возраста"

this->textBox18->Clear();

this->textBox18->Text = b.ToString();

}

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e) {

double d = Convert::ToDouble(this->textBox43->Text);

double el = Convert::ToDouble(this->textBox27->Text);

this->textBox16->Clear();

this->textBox16->Text = d.ToString();

double k = d \* el;//расчет лимита ответственности по страховому случаю "смерть"

this->textBox19->Clear();

this->textBox19->Text = k.ToString();

}

private: System::Void button6\_Click(System::Object^ sender, System::EventArgs^ e) {

double l = Convert::ToDouble(this->textBox44->Text);

double m = Convert::ToDouble(this->textBox28->Text);

this->textBox14->Clear();

this->textBox14->Text = l.ToString();

double r = l \* m;//расчет лимита ответственности по страховому случаю "инвалидность"

this->textBox20->Clear();

this->textBox20->Text = r.ToString();

}

private: System::Void button7\_Click(System::Object^ sender, System::EventArgs^ e) {

double v = Convert::ToDouble(this->textBox45->Text);

double kook = Convert::ToDouble(this->textBox29->Text);

this->textBox15->Clear();

this->textBox15->Text = v.ToString();

double vkook = v \* kook;//расчет лимита ответственности по страховому случаю "вред здоровью"

this->textBox21->Clear();

this->textBox21->Text = vkook.ToString();

}

private: System::Void button8\_Click(System::Object^ sender, System::EventArgs^ e) {

double yon = Convert::ToDouble(this->textBox46->Text);

double min = Convert::ToDouble(this->textBox30->Text);

this->textBox17->Clear();

this->textBox17->Text = yon.ToString();

double yonmin = yon \* min;//расчет лимита ответственности по страховому случаю "временная нетрудоспособность"

this->textBox22->Clear();

this->textBox22->Text = yonmin.ToString();

}

private: System::Void button11\_Click(System::Object^ sender, System::EventArgs^ e)

{//расчет страховой суммы

double jk = Convert::ToDouble(this->textBox18->Text);

double vmin = Convert::ToDouble(this->textBox19->Text);

double vkook = Convert::ToDouble(this->textBox20->Text);

double namjin = Convert::ToDouble(this->textBox21->Text);

double jhope = Convert::ToDouble(this->textBox22->Text);

double lms = jk + vmin + vkook + namjin + jhope;

this->textBox25->Clear();

this->textBox25->Text = lms.ToString();

}

private: System::Void button9\_Click(System::Object^ sender, System::EventArgs^ e)

{//расчет переодического взноса

double vа = Convert::ToDouble(this->textBox13->Text);

double rm = Convert::ToDouble(this->textBox16->Text);

double kooki = Convert::ToDouble(this->textBox14->Text);

double jm = Convert::ToDouble(this->textBox15->Text);

double jin = Convert::ToDouble(this->textBox17->Text);

double msh = vа + rm + kooki + jm + jin;

this->textBox23->Clear();

this->textBox23->Text = msh.ToString();

}

private: System::Void button12\_Click(System::Object^ sender, System::EventArgs^ e)

{//расчет налогового вычета

double vi = Convert::ToDouble(this->textBox24->Text);//общий взнос по договору

//double tm = Convert::ToDouble(this->textBox23->Text); //периодический взнос

double bt2 = (vi \* 13 ) / 100;

this->textBox41->Clear();

this->textBox41->Text = bt2.ToString();

}

private: System::Void button10\_Click(System::Object^ sender, System::EventArgs^ e)

{//расчет общего взноса по договору взависимости от выбранного срока договора

double v = Convert::ToDouble(this->textBox23->Text);//периодический взнос

double nm = Convert::ToDouble(this->comboBox4->Text);

double vn = Convert::ToDouble(this->comboBox5->Text);

if (nm == 3)

{

double d = 36 \* v;

this->textBox24->Clear();

this->textBox24->Text = d.ToString();

}

else if (nm == 3 && vn == 1)

{

double u = 37 \* v;

this->textBox24->Clear();

this->textBox24->Text = u.ToString();

}

else if (nm == 3 && vn == 2)

{

double y = 38 \* v;

this->textBox24->Clear();

this->textBox24->Text = y.ToString();

}

else if (nm == 3 && vn == 3)

{

double h = 39 \* v;

this->textBox24->Clear();

this->textBox24->Text = h.ToString();

}

else if (nm == 3 && vn == 4)

{

double g= 40 \* v;

this->textBox24->Clear();

this->textBox24->Text = g.ToString();

}

else if (nm == 3 && vn == 5)

{

double k = 41 \* v;

this->textBox24->Clear();

this->textBox24->Text = k.ToString();

}

else if (nm == 3 && vn == 6)

{

double l = 42 \* v;

this->textBox24->Clear();

this->textBox24->Text = l.ToString();

}

else if (nm == 3 && vn == 7)

{

double b = 43 \* v;

this->textBox24->Clear();

this->textBox24->Text = b.ToString();

}

else if (nm == 3 && vn == 9)

{

double j = 44 \* v;

this->textBox24->Clear();

this->textBox24->Text = j.ToString();

}

else if (nm == 3 && vn == 10)

{

double gm = 45 \* v;

this->textBox24->Clear();

this->textBox24->Text = gm.ToString();

}

else if (nm == 3 && vn == 11)

{

double dm = 46 \* v;

this->textBox24->Clear();

this->textBox24->Text = dm.ToString();

}

else if (nm == 4)

{

double kt = 48 \* v;

this->textBox24->Clear();

this->textBox24->Text = kt.ToString();

}

else if (nm == 4 && vn == 1)

{

double kn = 49 \* v;

this->textBox24->Clear();

this->textBox24->Text = kn.ToString();

}

else if (nm == 4 && vn == 2)

{

double rt = 50 \* v;

this->textBox24->Clear();

this->textBox24->Text = rt.ToString();

}

else if (nm == 4 && vn == 3)

{

double tf = 51 \* v;

this->textBox24->Clear();

this->textBox24->Text = tf.ToString();

}

else if (nm == 4 && vn == 4)

{

double f = 52 \* v;

this->textBox24->Clear();

this->textBox24->Text = f.ToString();

}

else if (nm == 4 && vn == 5)

{

double lo = 53 \* v;

this->textBox24->Clear();

this->textBox24->Text = lo.ToString();

}

else if (nm == 4 && vn == 6)

{

double jp = 54 \* v;

this->textBox24->Clear();

this->textBox24->Text = jp.ToString();

}

else if (nm == 4 && vn == 7)

{

double mj = 55 \* v;

this->textBox24->Clear();

this->textBox24->Text = mj.ToString();

}

else if (nm == 4 && vn == 8)

{

double xs = 56 \* v;

this->textBox24->Clear();

this->textBox24->Text = xs.ToString();

}

else if (nm == 4 && vn == 9)

{

double xx = 57 \* v;

this->textBox24->Clear();

this->textBox24->Text = xx.ToString();

}

else if (nm == 4 && vn == 10)

{

double pz = 58 \* v;

this->textBox24->Clear();

this->textBox24->Text = pz.ToString();

}

else if (nm == 4 && vn == 11)

{

double ki = 59 \* v;

this->textBox24->Clear();

this->textBox24->Text = ki.ToString();

}

else if (nm == 5)

{

double hd = 60 \* v;

this->textBox24->Clear();

this->textBox24->Text = hd.ToString();

}

else if (nm == 5 && vn == 1)

{

double bt = 61 \* v;

this->textBox24->Clear();

this->textBox24->Text = bt.ToString();

}

else if (nm == 5 && vn == 2)

{

double bp = 62 \* v;

this->textBox24->Clear();

this->textBox24->Text = bp.ToString();

}

else if (nm == 5 && vn == 3)

{

double ts = 63 \* v;

this->textBox24->Clear();

this->textBox24->Text = ts.ToString();

}

else if (nm == 5 && vn == 4)

{

double fr = 64 \* v;

this->textBox24->Clear();

this->textBox24->Text = fr.ToString();

}

else if (nm == 5 && vn == 5)

{

double gt = 65 \* v;

this->textBox24->Clear();

this->textBox24->Text = gt.ToString();

}

else if (nm == 5 && vn == 6)

{

double vy = 66 \* v;

this->textBox24->Clear();

this->textBox24->Text = vy.ToString();

}

else if (nm == 5 && vn == 7)

{

double hu = 67 \* v;

this->textBox24->Clear();

this->textBox24->Text = hu.ToString();

}

else if (nm == 5 && vn == 8)

{

double m = 68 \* v;

this->textBox24->Clear();

this->textBox24->Text = m.ToString();

}

else if (nm == 5 && vn == 9)

{

double o = 69 \* v;

this->textBox24->Clear();

this->textBox24->Text = o.ToString();

}

else if (nm == 5 && vn == 10)

{

double zh = 70 \* v;

this->textBox24->Clear();

this->textBox24->Text = zh.ToString();

}

else if (nm == 5 && vn == 11)

{

double n = 71 \* v;

this->textBox24->Clear();

this->textBox24->Text = n.ToString();

}

else if (nm == 6)

{

double og = 72 \* v;

this->textBox24->Clear();

this->textBox24->Text = og.ToString();

}

else if (nm == 6 && vn == 1)

{

double mn = 73 \* v;

this->textBox24->Clear();

this->textBox24->Text = mn.ToString();

}

else if (nm == 6 && vn == 2)

{

double ep = 74 \* v;

this->textBox24->Clear();

this->textBox24->Text = ep.ToString();

}

else if (nm == 6 && vn == 3)

{

double et = 75 \* v;

this->textBox24->Clear();

this->textBox24->Text = et.ToString();

}

else if (nm == 6 && vn == 4)

{

double eh = 76 \* v;

this->textBox24->Clear();

this->textBox24->Text = eh.ToString();

}

else if (nm == 6 && vn == 5)

{

double er = 77 \* v;

this->textBox24->Clear();

this->textBox24->Text = er.ToString();

}

else if (nm == 6 && vn == 6)

{

double ey = 78 \* v;

this->textBox24->Clear();

this->textBox24->Text = ey.ToString();

}

else if (nm == 6 && vn == 7)

{

double ew = 79 \* v;

this->textBox24->Clear();

this->textBox24->Text = ew.ToString();

}

else if (nm == 6 && vn == 8)

{

double es = 80 \* v;

this->textBox24->Clear();

this->textBox24->Text = es.ToString();

}

else if (nm == 6 && vn == 9)

{

double ed = 81 \* v;

this->textBox24->Clear();

this->textBox24->Text = ed.ToString();

}

else if (nm == 6 && vn == 10)

{

double ea = 82 \* v;

this->textBox24->Clear();

this->textBox24->Text = ea.ToString();

}

else if (nm == 6 && vn == 11)

{

double eq = 83 \* v;

this->textBox24->Clear();

this->textBox24->Text = eq.ToString();

}

else if (nm == 7)

{

double lol = 84 \* v;

this->textBox24->Clear();

this->textBox24->Text = lol.ToString();

}

else if (nm == 7 && vn == 1)

{

double ez = 85 \* v;

this->textBox24->Clear();

this->textBox24->Text = ez.ToString();

}

else if (nm == 7 && vn == 2)

{

double eb = 86 \* v;

this->textBox24->Clear();

this->textBox24->Text = eb.ToString();

}

else if (nm == 7 && vn == 3)

{

double emn = 87 \* v;

this->textBox24->Clear();

this->textBox24->Text = emn.ToString();

}

else if (nm == 7 && vn == 4)

{

double elk = 88 \* v;

this->textBox24->Clear();

this->textBox24->Text = elk.ToString();

}

else if (nm == 7 && vn == 5)

{

double bom = 89 \* v;

this->textBox24->Clear();

this->textBox24->Text = bom.ToString();

}

else if (nm == 7 && vn == 6)

{

double ugh = 90 \* v;

this->textBox24->Clear();

this->textBox24->Text = ugh.ToString();

}

else if (nm == 7 && vn == 7)

{

double ivi = 91 \* v;

this->textBox24->Clear();

this->textBox24->Text = ivi.ToString();

}

else if (nm == 7 && vn == 8)

{

double iva = 92 \* v;

this->textBox24->Clear();

this->textBox24->Text = iva.ToString();

}

else if (nm == 7 && vn == 9)

{

double ava = 93 \* v;

this->textBox24->Clear();

this->textBox24->Text = ava.ToString();

}

else if (nm == 7 && vn == 10)

{

double ida = 94 \* v;

this->textBox24->Clear();

this->textBox24->Text = ida.ToString();

}

else if (nm == 7 && vn == 11)

{

double ika = 95 \* v;

this->textBox24->Clear();

this->textBox24->Text = ika.ToString();

}

else if (nm == 8)

{

double sos = 96 \* v;

this->textBox24->Clear();

this->textBox24->Text = sos.ToString();

}

else if (nm == 8 && vn == 1)

{

double spa = 97 \* v;

this->textBox24->Clear();

this->textBox24->Text = spa.ToString();

}

else if (nm == 8 && vn == 2)

{

double siti = 98 \* v;

this->textBox24->Clear();

this->textBox24->Text = siti.ToString();

}

else if (nm == 8 && vn == 3)

{

double plz = 99 \* v;

this->textBox24->Clear();

this->textBox24->Text = plz.ToString();

}

else if (nm == 8 && vn == 4)

{

double help = 100 \* v;

this->textBox24->Clear();

this->textBox24->Text = help.ToString();

}

else if (nm == 8 && vn == 5)

{

double kill = 101 \* v;

this->textBox24->Clear();

this->textBox24->Text = kill.ToString();

}

else if (nm == 8 && vn == 6)

{

double lopuh = 102 \* v;

this->textBox24->Clear();

this->textBox24->Text = lopuh.ToString();

}

else if (nm == 8 && vn == 7)

{

double hochu = 103 \* v;

this->textBox24->Clear();

this->textBox24->Text = hochu.ToString();

}

else if (nm == 8 && vn == 8)

{

double go = 104 \* v;

this->textBox24->Clear();

this->textBox24->Text = go.ToString();

}

else if (nm == 8 && vn == 9)

{

double home = 105 \* v;

this->textBox24->Clear();

this->textBox24->Text = home.ToString();

}

else if (nm == 8 && vn == 10)

{

double fig = 106 \* v;

this->textBox24->Clear();

this->textBox24->Text = fig.ToString();

}

else if (nm == 8 && vn == 11)

{

double na = 107 \* v;

this->textBox24->Clear();

this->textBox24->Text = na.ToString();

}

else if (nm == 9)

{

double hee = 108 \* v;

this->textBox24->Clear();

this->textBox24->Text = hee.ToString();

}

else if (nm == 9 && vn == 1)

{

double llp = 109 \* v;

this->textBox24->Clear();

this->textBox24->Text = llp.ToString();

}

else if (nm == 9 && vn == 2)

{

double lpp = 110 \* v;

this->textBox24->Clear();

this->textBox24->Text = lpp.ToString();

}

else if (nm == 9 && vn == 3)

{

double lop = 111 \* v;

this->textBox24->Clear();

this->textBox24->Text = lop.ToString();

}

else if (nm == 9 && vn == 4)

{

double kol = 112 \* v;

this->textBox24->Clear();

this->textBox24->Text = kol.ToString();

}

else if (nm == 9 && vn == 5)

{

double koll = 113 \* v;

this->textBox24->Clear();

this->textBox24->Text = koll.ToString();

}

else if (nm == 9 && vn == 6)

{

double kkl = 114 \* v;

this->textBox24->Clear();

this->textBox24->Text = kkl.ToString();

}

else if (nm == 9 && vn == 7)

{

double copy = 115 \* v;

this->textBox24->Clear();

this->textBox24->Text = copy.ToString();

}

else if (nm == 9 && vn == 8)

{

double cat = 116 \* v;

this->textBox24->Clear();

this->textBox24->Text = cat.ToString();

}

else if (nm == 9 && vn == 9)

{

double dog = 117 \* v;

this->textBox24->Clear();

this->textBox24->Text = dog.ToString();

}

else if (nm == 9 && vn == 10)

{

double kot = 118 \* v;

this->textBox24->Clear();

this->textBox24->Text = kot.ToString();

}

else if (nm == 9 && vn == 11)

{

double ruby = 119 \* v;

this->textBox24->Clear();

this->textBox24->Text = ruby.ToString();

}

else if (nm == 10)

{

double gufy = 120 \* v;

this->textBox24->Clear();

this->textBox24->Text = gufy.ToString();

}

else if (nm == 10 && vn == 1)

{

double pufi = 121 \* v;

this->textBox24->Clear();

this->textBox24->Text = pufi.ToString();

}

else if (nm == 10 && vn == 2)

{

double puf = 122 \* v;

this->textBox24->Clear();

this->textBox24->Text = puf.ToString();

}

else if (nm == 10 && vn == 3)

{

double paf = 123 \* v;

this->textBox24->Clear();

this->textBox24->Text = paf.ToString();

}

else if (nm == 10 && vn == 4)

{

double sqwid = 124 \* v;

this->textBox24->Clear();

this->textBox24->Text = sqwid.ToString();

}

else if (nm == 10 && vn == 5)

{

double ward = 125 \* v;

this->textBox24->Clear();

this->textBox24->Text = ward.ToString();

}

else if (nm == 10 && vn == 6)

{

double spanch = 126 \* v;

this->textBox24->Clear();

this->textBox24->Text = spanch.ToString();

}

else if (nm == 10 && vn == 7)

{

double bob = 127 \* v;

this->textBox24->Clear();

this->textBox24->Text = bob.ToString();

}

else if (nm == 10 && vn == 8)

{

double ui = 128 \* v;

this->textBox24->Clear();

this->textBox24->Text = ui.ToString();

}

else if (nm == 10 && vn == 9)

{

double wang = 129 \* v;

this->textBox24->Clear();

this->textBox24->Text = wang.ToString();

}

else if (nm == 10 && vn == 10)

{

double aziza = 130 \* v;

this->textBox24->Clear();

this->textBox24->Text = aziza.ToString();

}

else if (nm == 10 && vn == 11)

{

double nani = 131 \* v;

this->textBox24->Clear();

this->textBox24->Text = nani.ToString();

}

}

private: System::Void темаToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

}

};

}

//MyForm3.h

#pragma once

namespace курсовая2курс {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

public ref class MyForm3 : public System::Windows::Forms::Form

{

public:

MyForm3(void)

{

InitializeComponent();

this->StartPosition = FormStartPosition::CenterScreen;

}

protected:

~MyForm3()

{

if (components)

{

delete components;

}

}

#pragma endregion

private: System::Void MyForm3\_Load(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void label2\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void label1\_Click(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void русскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Text = "Справка";

видToolStripMenuItem->Text = "Вид";

языкToolStripMenuItem->Text = "Язык";

русскийToolStripMenuItem->Text = "Русский";

английскийToolStripMenuItem->Text = "Английский";

белорусскйToolStripMenuItem->Text = "Белорусский";

label1->Text = "Данная программа производит расчет для накопительного";

label2->Text = "страхования, исходя из введенных данных.";

label3->Text = "Работник должен ввести";

label4->Text = "свое ФИО и ФИО клиента, также заполнить поля с необходимой,";

label5->Text = "для заключения договора и расчета страховки, информацией,";

label6->Text = "которую должен предоставить клиент. Далее должны заполняться ";

label7->Text = "поля со страховыми случаями,";

label8->Text = "после того, как клиент выбрал страховые случаи, работник вместе c";

label9->Text = "клиентом должны заполнить поля с суммой, которую клиент будет";

label10->Text = "платить за выбранные страховые случаи. После заполнения";

label11->Text = "необходимых полей будет расчитана сумма полной страховки.";

}

private: System::Void белорусскйToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Text = "Даведка";

видToolStripMenuItem->Text = "Выгляд";

языкToolStripMenuItem->Text = "Мова";

русскийToolStripMenuItem->Text = "Рускій";

английскийToolStripMenuItem->Text = "Англійскій";

белорусскйToolStripMenuItem->Text = "Беларускій";

label1->Text = "Дадзеная праграма вырабляе разлік для назапашвальнага";

label2->Text = "страхавання, зыходзячы з уведзеных дадзеных.";

label3->Text = "Работнік павінен ўвесці";

label4->Text = "сваё прозвішча, імя і Прозвішча, імя кліента, таксама запоўніць палі з неабходнай";

label5->Text = "для заключэння дагавора і разліку страхоўкі, інфармацыяй,";

label6->Text = "якую павінен даць кліент. Далей павінны запаўняцца ";

label7->Text = "поля са страхавымі выпадкамі,";

label8->Text = "пасля таго, як кліент абраў страхавыя выпадкі, работнік разам c";

label9->Text = "кліентам павінны запоўніць поля з сумай, якую кліент будзе";

label10->Text = "плаціць за выбраныя страхавыя выпадкі. Пасля запаўнення";

label11->Text = "неабходных палёў будзе разлічана сума поўнай страхоўкі.";

}

private: System::Void английскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

this->Text = "Reference";

видToolStripMenuItem->Text = "View";

языкToolStripMenuItem->Text = "Language";

русскийToolStripMenuItem->Text = "Russian";

английскийToolStripMenuItem->Text = "English";

белорусскйToolStripMenuItem->Text = "Belarusian";

label1->Text = "This program makes a calculation for cumulative";

label2->Text = "сinsurance based on the data entered.";

label3->Text = "The employee must enter";

label4->Text = "his full name and the client's full name, and also fill in the fields with the necessary";

label5->Text = "information that the client must provide to conclude";

label6->Text = "the contract and calculate insurance. Next, the fields with insurance ";

label7->Text = "cases must be filled in, after the client";

label8->Text = "has selected insurance cases, the employee together with";

label9->Text = " the client must fill in the fields with the amount ";

label10->Text = "that the client will pay for the selected insurance cases. After filling in the required";

label11->Text = "fields, the amount of full insurance will be calculated.";

}

private: System::Void label8\_Click(System::Object^ sender, System::EventArgs^ e) {

}

};

}

# ПРИЛОЖЕНИЕ Б

**(справочное)**

**результат работы программы**

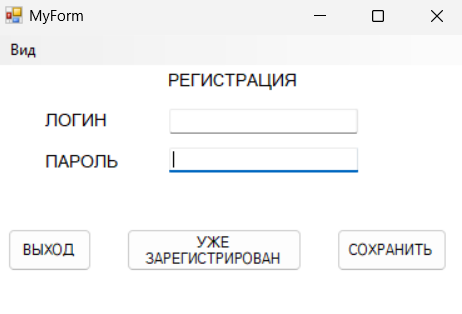


Рисунок Б. 1 – Результат функции «Запустить приложение»

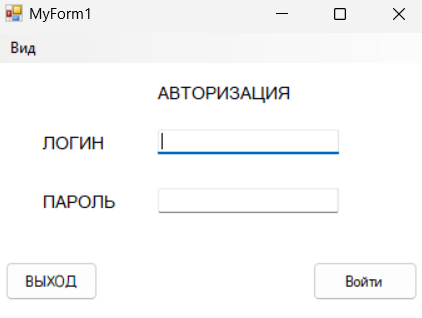


Рисунок Б. 2 – Результат функции «Регистрация»



Рисунок Б. 3 – Результат успешной регистрации

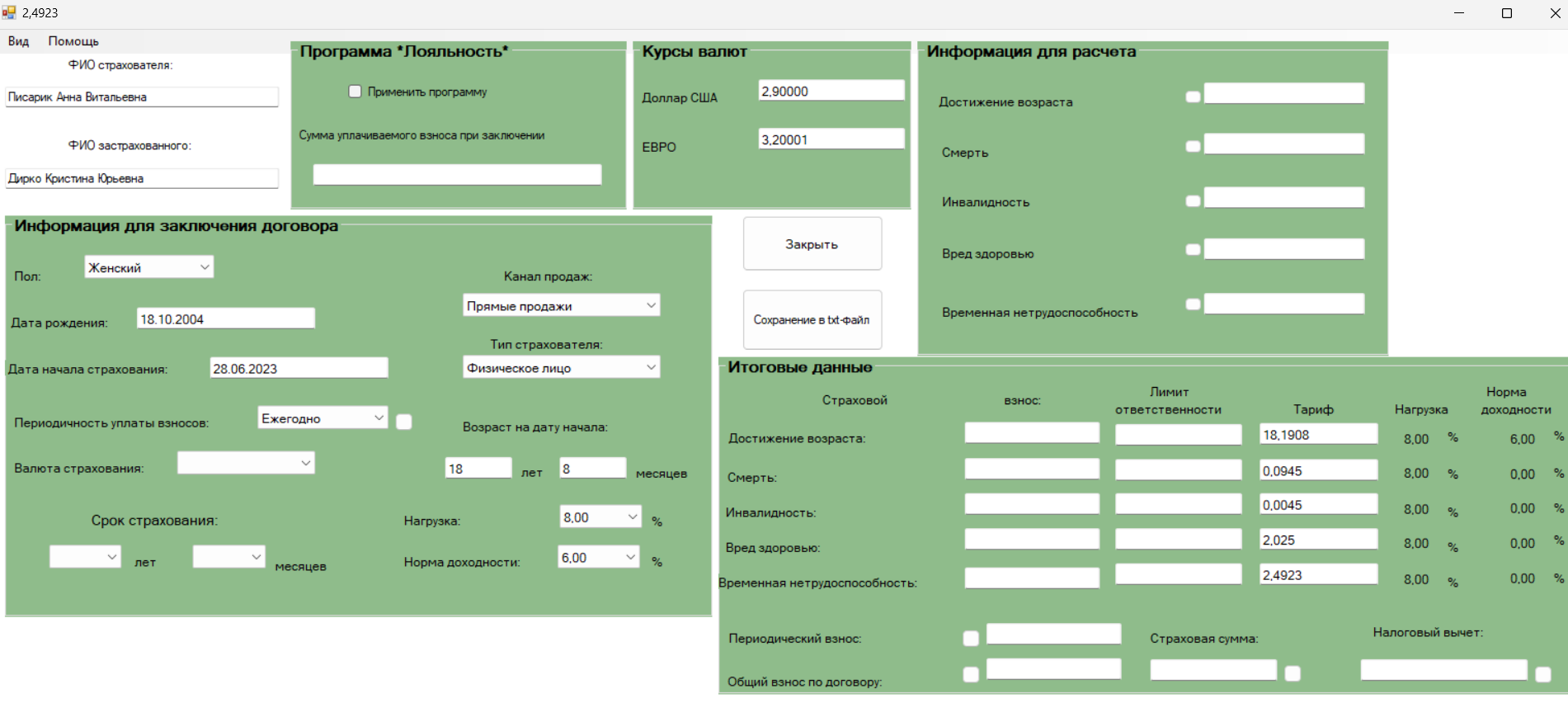


Рисунок Б. 4 – Результат работы функции «Периодичность уплаты взносов»

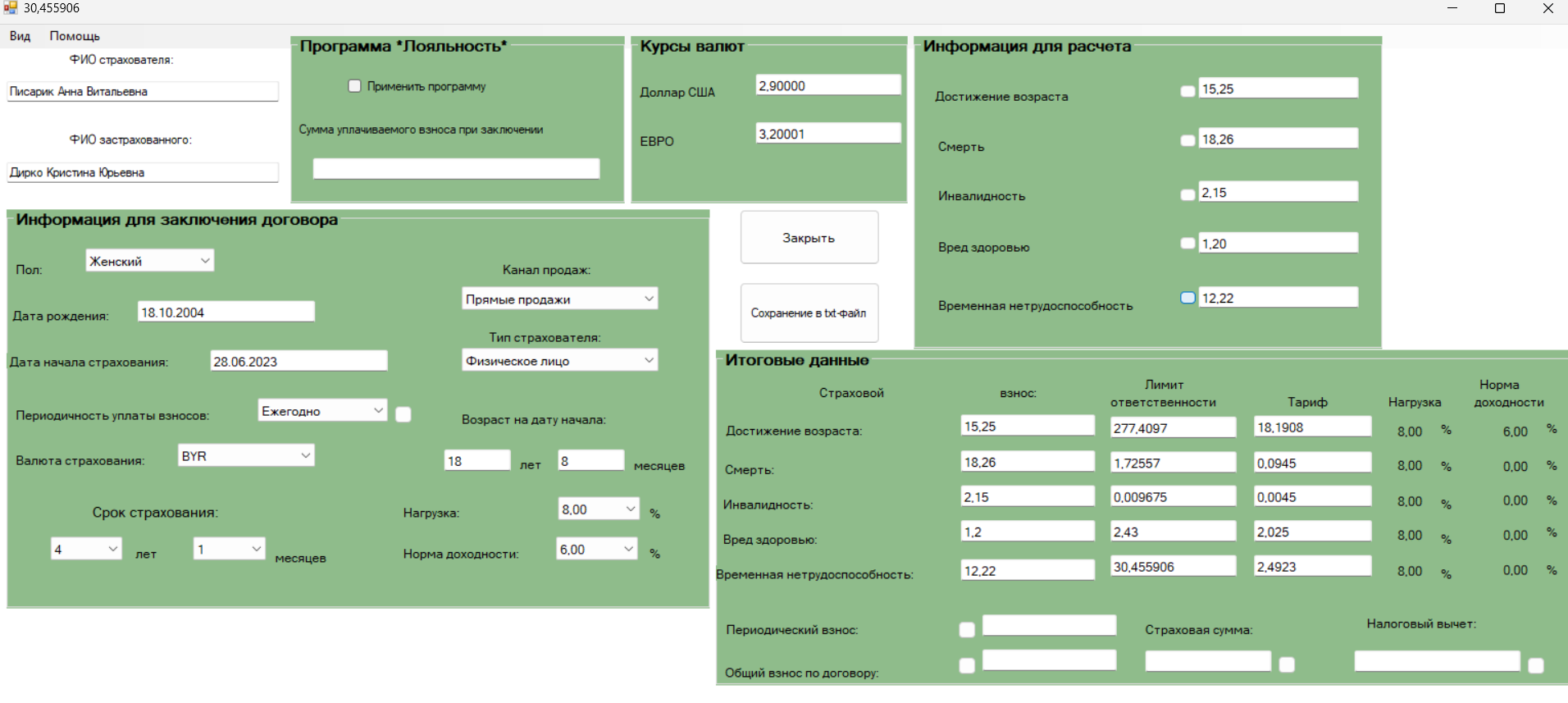


Рисунок Б. 5 – Результат работы функции «Лимит ответственности»

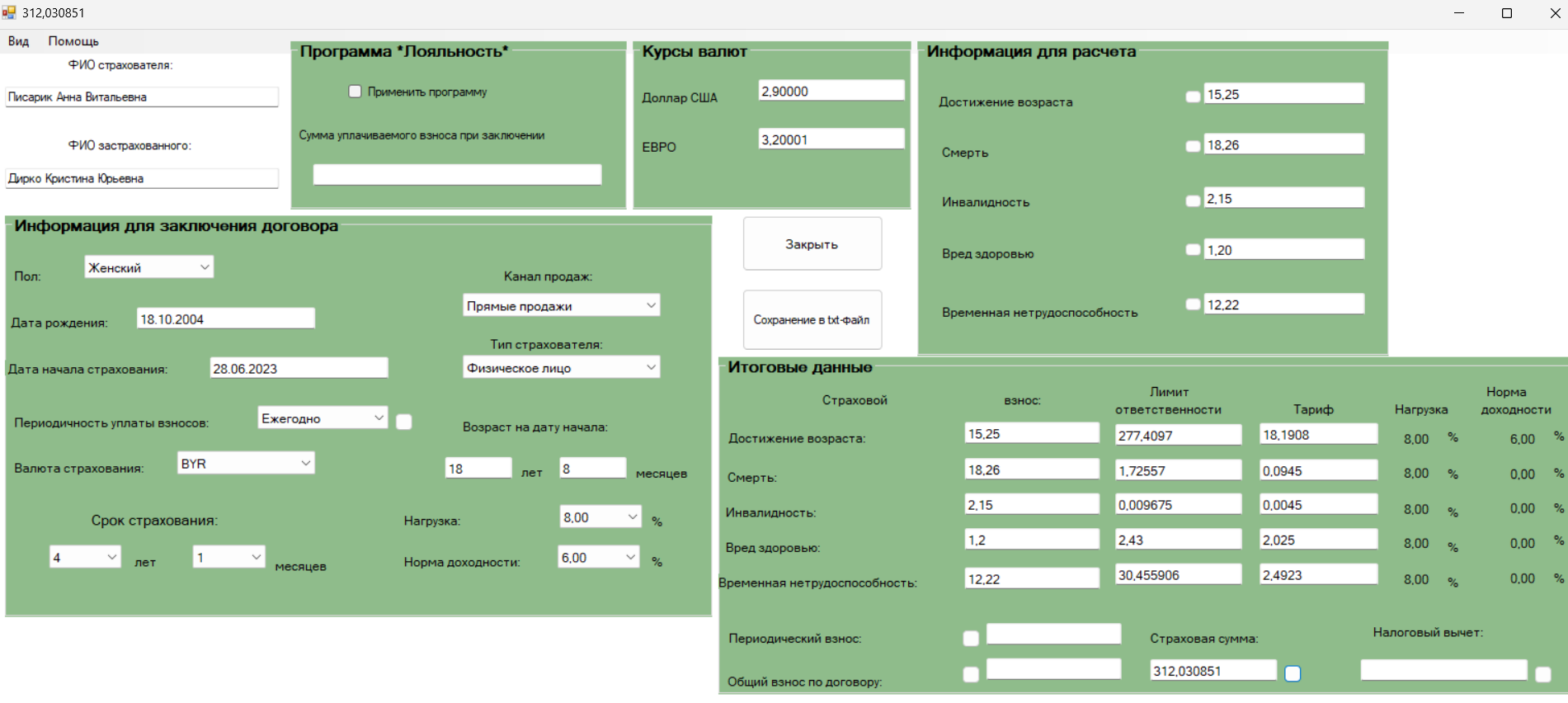


Рисунок Б. 6 – Результат работы функции «Страховая сумма»

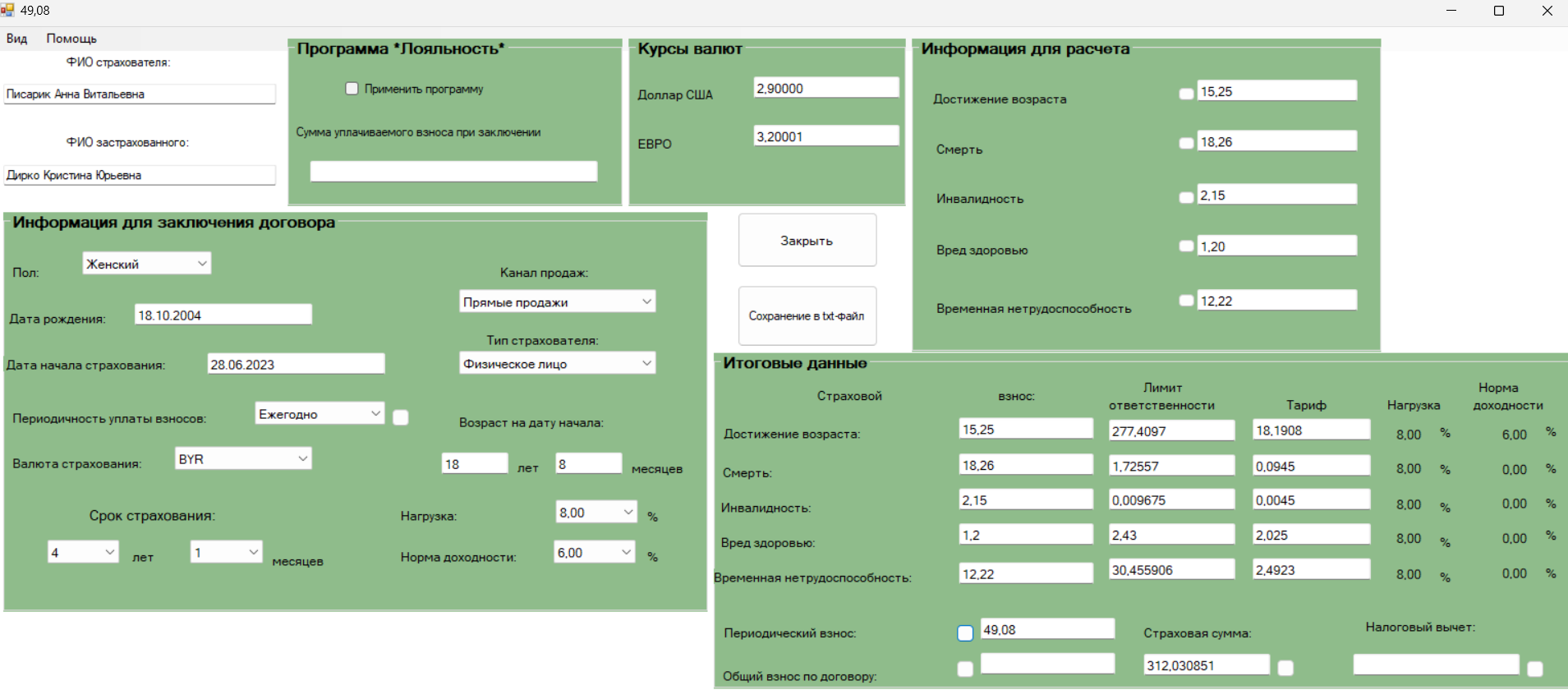


Рисунок Б. 7 – Результат работы функции «Периодический взнос»

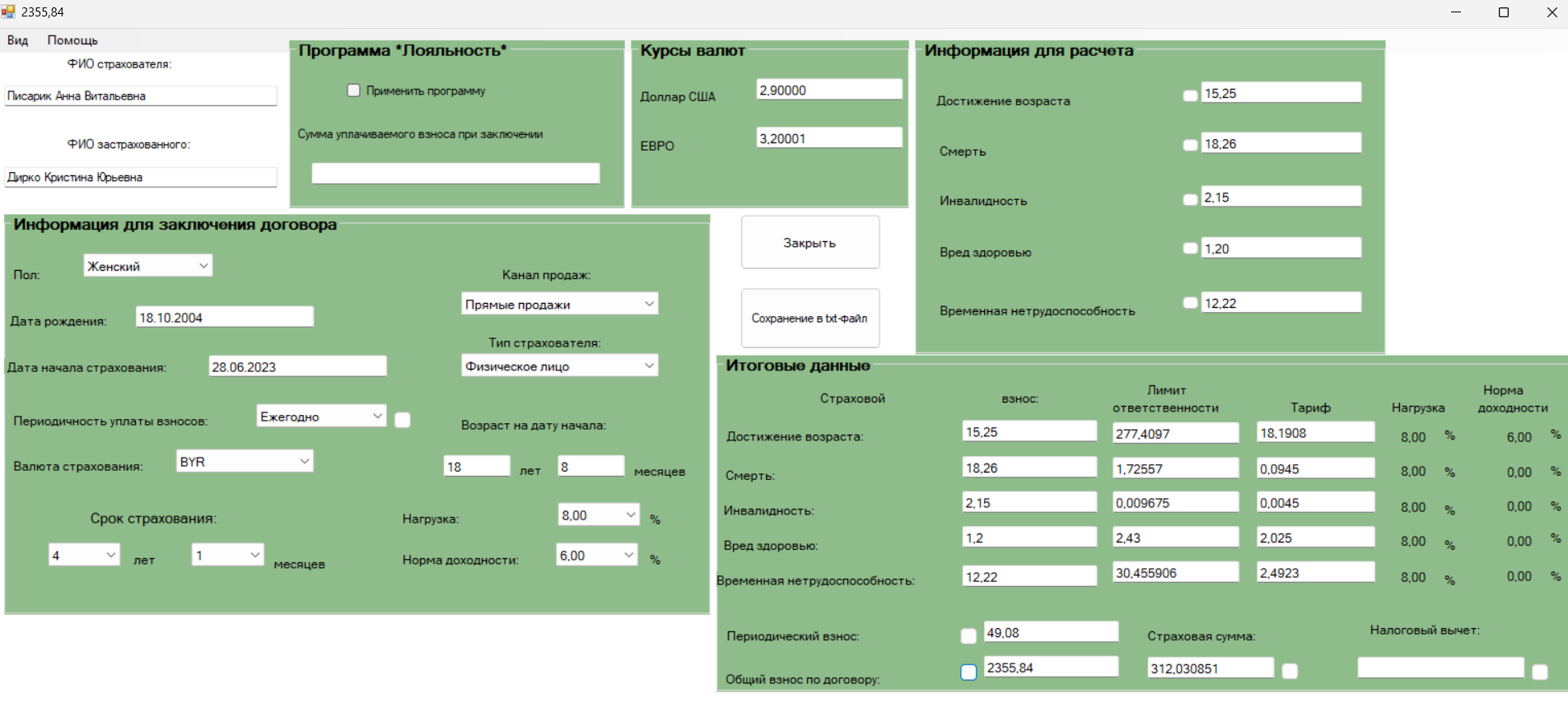


Рисунок Б. 8 – Результат функции «Общий взнос по договору»

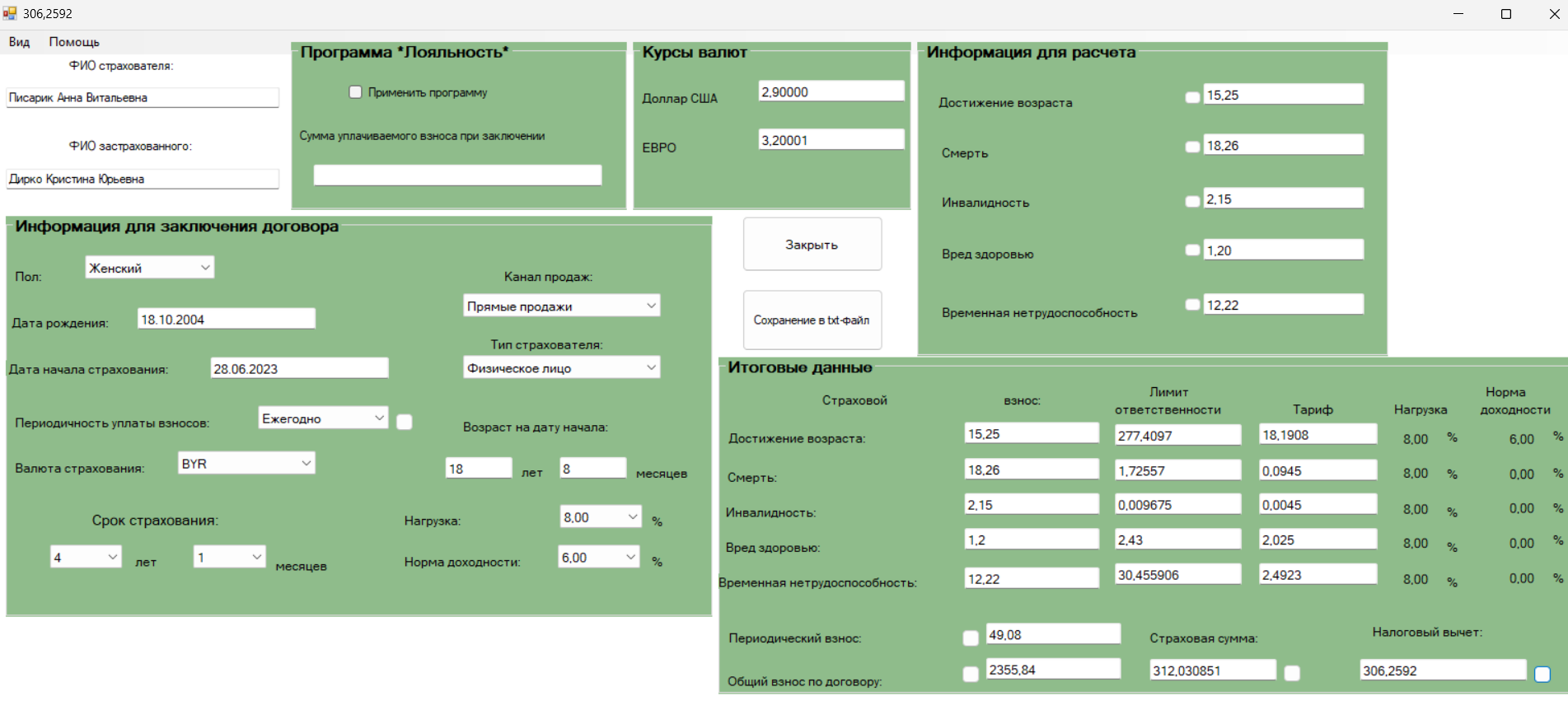


Рисунок Б. 9 – Результат функции «Налоговый вычет»



КП Т.219019.401 ГЧ

КБП

У

Т. Контр.

Разраб. Разраб.

Писарик А.В.

Провер. Провер.

Шаляпин Ю.В.

Реценз.

Утверд.

Н. Контр.

Программное средство для расчета страховых накопительных выплат

Блок-схема работы программы

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 2

Листов 2

Инв.№подл.

Подп. и дата

Взам.инв.№

Инв.№дубл.

Подп. и дата

КП Т.219019.401 ГЧ

КП Т.219019.401 ГЧ

КБП

У

Т. Контр.

Разраб. Разраб.

Писарик А.В.

Провер. Провер.

Шаляпин Ю.В.

Реценз.

Утверд.

Н. Контр.

Программное средство для расчета страховых накопительных выплат

Блок-схема работы одного из алгоритмов вычисления функции

№ докум.

Подпись

Дата

Изм.

Лист

Масса

Лит.

Масштаб

Лист 2

Листов 2

Инв.№подл.

Подп. и дата

Взам.инв.№

Инв.№дубл.

Подп. и дата

КП Т.219019.401 ГЧ



**Удостоверяющий лист**

электронного документа – курсовой проект

Тема «Программное средство для расчета страховых накопительных выплат».

Обозначение \_\_\_\_\_\_\_\_\_КП Т.219019.401\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Разработчик \_Писарик А.В.\_\_\_ Руководитель Шаляпин\_Ю.В.\_\_

(Ф.И.О.) (Ф.И.О.)

Подписи лиц, ответственных за разработку электронного документа

|  |  |  |
| --- | --- | --- |
| Состав электронного документа | Разработчик | Руководитель |
| Пояснительная записка (на бумажном носителе формата А4), Пояснительная записка.docх |  |  |
| Папка с проектом «Курсовая\_2курс» |  |  |
| ГЧ, Блок-схема работы программы(общая).vsd |  |  |
| ГЧ, Блок-схема работы алгоритма расчета периодического взноса.vsd |  |  |
| Тип носителя: USB-флеш-накопитель |  |  |

**Этикетка**

для курсовых проектов

**Курсовой проект**

Тема «Программное средство для расчета страховых накопительных выплат»

КП Т.219019.401

Разработан

Утвержден

Разработчик Писарик А.В.

Руководитель: Ю.В. Шаляпин

Технические средства: Компьютер с процессором Intel Core i7 – 1165G7 от 2.8 ГГц, ОЗУ от 4 ГБ, 500 МБ свободного места на диске, монитор, клавиатура, мышка

Программные средства: Операционная система MS Windows 11 Home, MS Visual Studio 2022, MS Visio 2007, текстовый редактор Microsoft Word 2019, Notepad.exe

**Состав документа:**

Пояснительная записка – Пояснительная записка.docх

Графическая часть – Блок-схема работы алгоритма расчета периодического взноса.docx, Блок-схема работы программы (общая).docx

Папка с проектом – Курсовая\_2курс

Сведения о защите информации: пароль отсутствует