**РЕФЕРАТ**

Дипломный проект содержит 139 страницы, 34 рисунков, 10 таблиц, 1 приложение. Библиографический список включает 21 публикацию.

ПРОГНОЗИРОВАНИЕ, АНАЛИЗ СТРАХОВОГО ПОРТФЕЛЯ, АНДЕРРАЙТИНГ

*Цель проекта* **–** повышение точности прогнозирования портфеля автострахования посредством разработки программного обеспечения, основанного на применении методов анализа данных.

*Средства разработки* – среда программирования MS Visual Studio 2013 Ultimate, графическая подсистема Windows Presentation Foundation, язык программирования С#.

*Назначение разработанных средств* – автоматизация прогнозирования портфеля автострахования.

*Область применения* – бизнес-аналитика и страховые компании.

**ABSTRACT**

The degree project contains 139 pages, 34 figures, 10 tables, 1 addendum. Bibliographic list includes 21 publications.

FORECASTING, INSURANCE PORTFOLIO ANALYSIS, UNDERWRITING

*The purpose of the project* – the forecasting accuracy increasing of automobile insurance portfolio by means of the software design based on the application of data analyze algorithms.

*Purpose of the developed means* – the process automation of automobile insurance portfolio.

*The means of development* – the programming environment is MS Visual Studio 2013 Ultimate; graphics system is Windows Presentation Foundation; the programming language is C#.

*The scope* – business analytics and insurance companies.

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 9](#_Toc389545694)

[1. ПОСТАНОВКА ЗАДАЧИ 12](#_Toc389545695)

[2. ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА 14](#_Toc389545696)

[2.1. ОБОСНОВАНИЕ ПО ПРОВЕДЕНИЮ РАБОТ ПО РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 14](#_Toc389545697)

[2.2 ОБОСНОВАНИЕ ВЫБОРА СРЕДСТВ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 15](#_Toc389545698)

[3. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 18](#_Toc389545699)

[3.1. ПОНЯТИЕ СТРАХОВАНИЯ 18](#_Toc389545700)

[3.2. ИСТОРИЯ РАЗВИТИЯ АВТОТРАНСПОРТНОГО СТРАХОВАНИЯ И ЕГО СОВРЕМЕННАЯ ОЦЕНКА 25](#_Toc389545701)

[3.3. ВЗАИМОДЕЙСТВИЕ УЧАСТНИКОВ СТРАХОВЫХ ОТНОШЕНИЙ 28](#_Toc389545702)

[3.4. ОБЪЕМ СТРАХОВОЙ ОТВЕТСТВЕННОСТИ. ВАРИАНТЫ СТРАХОВАНИЯ 29](#_Toc389545703)

[3.5. МЕТОДИКА РАСЧЕТОВ СТРАХОВЫХ ТАРИФОВ. СТРУКТУРА ТАРИФНОЙ СТАВКИ 30](#_Toc389545704)

[3.6. СУЩНОСТЬ И ВИДЫ СТРАХОВОГО ПОРТФЕЛЯ 34](#_Toc389545705)

[4. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 42](#_Toc389545706)

[4.1. КЛАСТЕРИЗАЦИЯ СТРУКТУРЫ ПОРТФЕЛЯ АВТОТРАНСПОРТНОГО СТРАХОВАНИЯ ФИЗИЧЕСКИХ ЛИЦ 42](#_Toc389545707)

[4.2. АЛГОРИТМ K-СРЕДНИХ 43](#_Toc389545708)

[4.3. АЛГОРИТМ CLOPE 46](#_Toc389545709)

[4.4. ОПИСАНИЕ ЯЗЫКА И СРЕДСТВ ПРОГРАММИРОВАНИЯ 50](#_Toc389545710)

[**4.4.1. ЯЗЫК ПРОГРАММИРОВАНИЯ C#** 50](#_Toc389545711)

[**4.4.2. ОСНОВНЫЕ ТИПЫ WEB-САЙТОВ** 51](#_Toc389545712)

[5. ПРОЕКТНАЯ ЧАСТЬ 53](#_Toc389545713)

[5.1. ДИАГРАММА ПРЕЦЕДЕНТОВ ИСПОЛЬЗОВАНИЯ 54](#_Toc389545714)

[5.2. ДИАГРАММА КЛАССОВ 56](#_Toc389545715)

[**5.2.1. ОПИСАНИЕ КЛАССОВ** 57](#_Toc389545716)

[**5.2.2. ОПИСАНИЕ ОТНОШЕНИЙ МЕЖДУ КЛАССАМИ** 60](#_Toc389545717)

[5.3. ДИАГРАММА ВЗАИМОДЕЙСТВИЯ ОБЪЕКТОВ 61](#_Toc389545718)

[5.4. ДИАГРАММА ДЕЯТЕЛЬНОСТИ 61](#_Toc389545719)

[5.5 ДИАГРАММА КОМПОНЕНТОВ 63](#_Toc389545720)

[6. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 66](#_Toc389545721)

[6.1 ОБЩИЙ АЛГОРИТМ РАБОТЫ ПРОГРАММЫ 66](#_Toc389545722)

[6.2. СОСТАВ ПРОЕКТА 66](#_Toc389545723)

[6.3 СПЕЦИФИКАЦИЯ ВНУТРЕННИХ ФУНКЦИЙ 68](#_Toc389545724)

[6.4 СПЕЦИФИКАЦИЯ ДАННЫХ 69](#_Toc389545725)

[7 ПРОГРАММНАЯ ДОКУМЕНТАЦИЯ 71](#_Toc389545726)

[7.1 НАЗНАЧЕНИЕ ПРОГРАММЫ 71](#_Toc389545727)

[7.2 Требования к составу и параметрам технических средств 71](#_Toc389545728)

[7.3 Руководство пользователя 72](#_Toc389545729)

[7.4 Руководство программиста 80](#_Toc389545730)

[8 ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 81](#_Toc389545731)

[8.1 ОБЪЕКТ ИСПЫТАНИЙ 81](#_Toc389545732)

[8.2 ЦЕЛЬ ИСПЫТАНИЙ 81](#_Toc389545733)

[8.3 ТРЕБОВАНИЯ К ПРОГРАММЕ 81](#_Toc389545734)

[8.4 СРЕДСТВА ИСПЫТАНИЙ 81](#_Toc389545735)

[8.5 ПОРЯДОК ИСПЫТАНИЙ 82](#_Toc389545736)

[8.6 ИСПЫТАНИЯ ПРОГРАММЫ 82](#_Toc389545737)

[9 ЭКОНОМИЧЕСКАЯ ЧАСТЬ 86](#_Toc389545738)

[9.1. ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ 86](#_Toc389545739)

[9.2 ПЛАНИРОВАНИЕ ПОЭТАПНОЙ РАЗРАБОТКИ ПО 87](#_Toc389545740)

[9.3 СОСТАВЛЕНИЕ СМЕТЫ ЗАТРАТ НА РАЗРАБОТКУ ПО 91](#_Toc389545741)

[**9.3.1 МАТЕРИАЛЬНЫЕ ЗАТРАТЫ** 91](#_Toc389545742)

[**9.3.2 ЗАТРАТЫ НА ОПЛАТУ ТРУДА** 92](#_Toc389545743)

[**9.3.3 АМОРТИЗАЦИОННЫЕ ОТЧИСЛЕНИЯ** 93](#_Toc389545744)

[**9.3.4 ПРОЧИЕ РАСХОДЫ** 94](#_Toc389545745)

[**9.3.5 РАСЧЕТ ЦЕНЫ НИР** 95](#_Toc389545746)

[9.4 ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ 96](#_Toc389545747)

[10 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ ПРОЕКТА 99](#_Toc389545748)

[10.1 ОПИСАНИЕ РАБОЧЕГО ПОМЕЩЕНИЯ, РАБОЧЕГО МЕСТА, ВЫПОЛНЯЕМЫХ ОПЕРАЦИЙ 99](#_Toc389545749)

[10.2 АНАЛИЗ ОПАСНЫХ И ВРЕДНЫХ ФАКТОРОВ, ВОЗДЕЙСТВУЮЩИХ НА ПОЛЬЗОВАТЕЛЯ ПЭВМ 103](#_Toc389545750)

[10.3 ВОЗДЕЙСТВИЕ ПРОИЗВОДСТВЕННЫХ ФАКТОРОВ 105](#_Toc389545751)

[10.4 МЕРОПРИЯТИЯ ПО РАЗРАБОТКЕ БЕЗОПАСНЫХ УСЛОВИЙ ТРУДА НА РАБОЧЕМ МЕСТЕ 108](#_Toc389545752)

[10.5 ОБЕСПЕЧЕНИЕ ЭЛЕКТРОБЕЗОПАСНОСТИ НА РАБОЧЕМ МЕСТЕ 108](#_Toc389545753)

[10.6 ОБЕСПЕЧЕНИЕ ПОЖАРНОЙ БЕЗОППАСНОСТИ НА РАБОЧЕМ МЕСТЕ 110](#_Toc389545754)

[10.7 ИНЖЕНЕРНЫЕ РАСЧЕТЫ 112](#_Toc389545755)

[10.8 ЭКОЛОГИЧНОСТЬ 114](#_Toc389545756)

[ЗАКЛЮЧЕНИЕ 116](#_Toc389545757)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 118](#_Toc389545758)

[ПРИЛОЖЕНИЕ 120](#_Toc389545759)

**ВВЕДЕНИЕ**

Сфера деятельности всех страховых компаний – это коммерческое страхование, заключающегося в обеспечении договора страхования между страховой компанией и страхователем, в соответствии с условиями которого компания обязуется компенсировать ущерб в той или иной форме либо выплатить страхователю определенную денежную сумму при наступлении предусмотренного договором страхового случая – события, при наступлении которого возникает обязанность страховщика осуществить выплату страхователю. Объектом страхования может выступать различное имущество, в том числе и автомобили.

За предоставление услуг страхования страховщик выплачивает компании (так называемую страховую премию), единовременно или частями, пропорционально всему сроку страхования, предусмотренному договором.

Таким образом, главной задачей страховой компании является определение размера страховой премии, на основании ряда показателей и дальнейшее управление страховыми тарифами.

Целью дипломного проекта является повышение точности прогнозирования портфеля автострахования посредством разработки программного обеспечения, основанного на применении методов анализа данных.

На данный момент на рынке существуют лишь некоторые методики решения задачи анализа страхового портфеля, реализуемые лишь в специальных статистических программах (например, таких как Statistica), но комплексное специализируемое программное обеспечение отсутствует, что является основанием для разработки данного программного продукта.

Пояснительная записка дипломного проекта состоит из следующих десяти разделов.

1. Постановка задачи. В данном разделе указывается цель дипломного проекта, задачи, принципы и требования разработки ПО, а так же требования к системным ресурсам и средства разработки программного обеспечения.
2. Технико-экономическое обоснование. В данном разделе описаны характеристики ПО и его особенности. Так же дано обоснование разработки ПО с экономической точки зрения.
3. Анализ предметной области. В данном разделе содержится описание предметной области и ее особенностей.
4. Теоретическая часть. В данном разделе содержится описание выбранных инструментов анализа данных, а также языка и среды программирования.
5. Проектная часть. В данном разделе описывается разработка ПО с помощью языка моделирования UML.
6. Программная реализация. В данном разделе описана разработка логической структуры и непосредственная реализация ПО.
7. Программная документация. В данном разделе содержится описание применения ПО (назначение и условия его применения), руководство системного программиста, в котором указываются сведения о настройке ПО, и руководство пользователя.
8. Тестирование программного обеспечения. В данном разделе содержится описание проверки корректности работы ПО, а так же результаты тестирования и их анализ.
9. Экономическая часть. В данном разделе содержится расчет затрат на разработку ПО. Приведены выводы по эффективности использования программного обеспечения для автоматизации прогнозирования показателей страхового портфеля.
10. Безопасность и экологичность проекта. В данном разделе описаны организация рабочего места при эксплуатации ПЭВМ и обеспечение пожарной безопасности.

В приложении приведен прокомментированный листинг наиболее значимых частей ПО.

В качестве среды разработки выбрана Microsoft Visual Studio 2013 Ultimate, графическая подсистема Windows Presentation Foundation, язык программирования C#.

1. **ПОСТАНОВКА ЗАДАЧИ**

Целью данного дипломного проекта является повышение точности прогнозирования портфеля автострахования посредством разработки программного обеспечения, основанного на применении методов анализа данных.

ПО должно осуществлять главную цель – автоматизировать процесс анализа структуры страхового портфеля на основе заключенных договоров с известными параметрами в прошлом.

Для создания ПО необходимо определиться с алгоритмами анализа данных, которые будут реализовываться на программном уровне, и адекватно решать поставленную задачу.

По результатам сравнительного анализа алгоритмов, применяемых при решении различных задач анализа данных, было принято решение об использовании алгоритма CLOPE, чьи высокая масштабируемость и скорость работы, за счет минимально возможного количества "сканирований" объема анализируемых данных, позволяют получить результаты анализа показателей в кратчайшие сроки. Другим используемым методом является алгоритм k-средних, достоинствами которого являются быстрота и простота реализации.

Для достижения поставленной цели ПО с практической точки зрения должно удовлетворять следующим основным характеристикам.

1. ПО должно обеспечивать считывание исходных данных из файла (информацию достаточно большого объема неудобно и не эффективно вносить вручную);
2. ПО должно осуществлять анализ с помощью алгоритмов;
3. ПО должно обеспечить максимально возможный уровень точности найденных прогнозных значений;
4. ПО должно сохранять полученные результаты в рабочей книге Excel и обеспечивать возможность вывода их на печать.

Требования к разрабатываемому программному обеспечению

1. Работа ПО должна обеспечиваться на персональной ЭВМ, построенной на платформе Intel или AMD. К таким ПЭВМ относят различные модификации PC/AT с процессорами Intel или Athlon.
2. ПО должно обрабатывать заданные объемы информации при отсутствии сбоев.
3. ПО должно иметь удобный для пользователя интерфейс, и тем самым обеспечивать простоту восприятия и работы.
4. В ПО должен быть предусмотрен контроль правильности вводимых пользователем данных. В случае некорректного ввода данных пользователю должны выдаваться соответствующие диагностические сообщения. При этом некорректный ввод данных никак не должен отражаться на работоспособности.
5. ПО должно обладать достаточным быстродействием.

Для нормальной работы необходимо, чтобы конфигурация компьютера соответствовала требованиям, предъявляемым операционной системой к системным ресурсам. ПО разработано для операционных систем семейства Windows: Windows XP/Windows 7/ Windows 8, с установленным .NET Framework 3,5. Минимальной конфигурации оборудования соответствует процессор с частотой не ниже 1 ГГц, оперативная память объемом не менее 1 Гб и Видео адаптер с поддержкой DirectX 9. Но для более корректной и быстрой работы рекомендуется использовать процессор с частотой 2.1 ГГц и оперативную память объемом 2 Гб.

1. **ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ ПРОЕКТА**
   1. **ОБОСНОВАНИЕ ПО ПРОВЕДЕНИЮ РАБОТ ПО РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Прогнозирование и анализ является важной частью повседневной человеческой деятельности. Не в последнюю очередь это касается и рынка страхования. Задачей первостепенной важности для всех специалистов рынка автострахования является получение адекватных прогнозов об изменениях показателей портфеля автотранспортного страхования физических лиц в случаях изменения его структуры, для своевременного принятия решения об использовании подходящей стратегии управления тарифной политикой. Программные продукты данного направления должны обладать известным уровнем автоматизации, чтобы их пользователи могли сконцентрироваться на заключении договоров страхования, а не на анализе показателей, занимающем время, так как это может привести к упущению времени, вследствие этого к ошибкам, и, в худшем случае, к убыткам.

Одним из главных вопросов при разработке подобного программного обеспечения является выбор алгоритмов анализа. В данной работе использован алгоритм CLOPE, чьи высокая масштабируемость и скорость работы, за счет минимально возможного количества "сканирований" объема анализируемых данных, позволяют получить результаты анализа показателей в кратчайшие сроки. Другим используемым алгоритмом является алгоритм k-средних, достоинствами которого являются быстрота и простота реализации.

На данный момент на рынке существуют лишь некоторые методики решения задачи анализа страхового портфеля, реализуемые лишь в специальных статистических программах (например, таких как Statistica), но комплексное специализируемое программное обеспечение отсутствует, что является основанием для разработки данного программного продукта.

Разработка данного программного обеспечения, имеющего понятный интерфейс и спроектированного с точки зрения простоты использования конечным пользователем, позволит ускорить процесс обработки информации, обеспечит повышение производительности труда специалистов рынка страхования, сделает процесс анализа более простым и наглядным.

Таким образом, данная разработка является необходимой для специалистов рынка страхования, поскольку она автоматизирует процесс анализа структуры портфеля автотранспортного страхования физических лиц, уменьшая время до принятия решения по управлению тарифной политикой, и, следовательно, является экономически обоснованной.

* 1. **ОБОСНОВАНИЕ ВЫБОРА СРЕДСТВ  
     РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

Наиболее удобной программной средой для разработки программного обеспечения является среда программирования MS Visual Studio 2013 Ultimate.

В настоящее время именно MS Visual Studio 2013 Ultimate находит большое применение в решении различных прикладных задач, так как представляет собой эффективную среду профессионального разработчика, соединяющую несколько технологий, предоставляющих разработчикам широкие возможности для создания приложений Windows, Web-приложений, Web-служб, сайтов и т.д.

Данное средство создания приложений обладает рядом преимуществ:

* имеет язык С# в своей основе, что позволяет успешно реализовывать сложные для других языков алгоритмы и методы удобными и простыми способами;
* снабжено широким спектром методов и средств для разработки интуитивно-понятного графического интерфейса пользователя.
* обеспечено широкими возможностями по реализации работы с базами данных;

Язык высокого уровня С# содержит ряд важных новшеств, поднимающих искусство программирования на новую ступень. Например, в состав элементов языка С# включены такие понятия, как делегаты (представители), свойства, индексаторы и события. Подобно Java, язык С# предлагает средства динамического обнаружения ошибок, обеспечения безопасности и управляемого выполнения программ. Но, в отличие от Java, C# дает программистам доступ к указателям.

Таким образом, С# сочетает возможности C++ с типовой безопасностью Java, которая обеспечивается наличием механизма контроля типов (type checking) и корректным использованием шаблонных классов (template class). Более того, язык С# отличается тем, что компромисс между мощью и надежностью тщательно сбалансирован и практически прозрачен (не заметен для пользователя или программы).

Стоит отметить, что MS Visual Studio 2013 Ultimate является более понятной и гибкой средой разработки, чем ее аналоги (C++ Builder и Visual C++). Это позволяет быстро и грамотно реализовывать поставленные задачи. Кроме того, MS Visual Studio 2013 обладает удобными инструментами для построения графического интерфейса пользователя.

В тоже время интегрированная среда разработки MS Visual Studio 2013 Ultimate имеет в своем составе отладчик, предоставляющий широкие возможности для поиска и устранения ошибок, что существенно облегчает задачу разработчика и позволяет разработать приложение более грамотно и быстро.

Таким образом, использование в данной работе среды программирования MS Visual Studio 2013 Ultimate, располагающей всеми необходимыми средствами для применения алгоритмов анализа данных, и дает очевидные преимущества в вопросе точности и объективности полученного результата.

1. **АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

**3.1 ПОНЯТИЕ СТРАХОВАНИЯ**

Страхование - одна из древнейших категорий общественных отношений. Зародившись в период разложения первобытнообщинного строя, оно постепенно стало непременным спутником общественного производства. Первоначальный смысл рассматриваемого понятия связан со словом "страх". Владельцы имущества, вступая между собой в производственный отношения, испытывали страх за его сохранность, за возможность уничтожения или утраты в связи со стихийными бедствиями, пожарами, грабежами и другими непредвиденными опасностями экономической жизни [12].

Рискованный характер общественного производства − главная причина беспокойства каждого собственника имущества и товаропроизводителя за свое материальное благополучие. На этой почве закономерно возникла идея возмещения материального ущерба путем солидарной его раскладки между заинтересованными владельцами имущества. Если бы каждый отдельно взятый собственник попытался возместить ущерб за свой счет, то он был бы вынужден создавать материальные и денежные резервы, равные по величине стоимости своего имущества, что естественно разорительно.

Между тем жизненный опыт, основанный на многолетних наблюдениях, позволил сделать вывод о случайном характере наступления чрезвычайных событий и неравномерности нанесения ущерба. Было замечено, что число заинтересованных хозяйств часто бывает больше пострадавших от различных опасностей. При таких условиях солидарная раскладка ущерба между заинтересованными хозяйствами заметно сглаживает последствия стихий и других случайностей. При этом чем больше количество предприятий участвует в раскладке ущерба, тем меньшая доля средств приходится на одного участника. Так возникло страхование, сущность которого составляет замкнутая раскладка (имущества) ущерба.

Наиболее примитивной формой раскладки ущерба было натуральное страхование. За счет запаса зерна, фуража и других однородных, легко делимых продуктов, формируемых путем подушных натуральных взносов, оказывалась натуральная помощь пострадавшим крестьянским хозяйствам. Однако такое страхование ограничивалось естественными рамками однородности и делимости, формируемых с его помощью натуральных запасов, поэтому по мере развития товарно-денежных отношений оно уступило место страхованию в денежной форме.

Раскладка ущерба в денежной форме создавало широкие возможности, прежде всего для взаимного страхования, когда сумма ущерба возмещалась его участниками на солидарных началах либо после каждого страхового случая, либо по окончанию хозяйственного года. Взаимное страхование в условиях капитализма стало закономерно перерастать в самостоятельную отрасль страхового дела. Если при взаимном страховании еще не формировался заранее рассчитанный с помощью теории вероятности страховой фонд, то в дальнейшем вероятная средняя величина возможного ущерба, приходящегося на каждого участника страхования, стала применяться в качестве основы страховых взносов для заблаговременного формирования страхового фонда. В условиях современного общества страхование превратилось во всеобщее универсальное средство страховой защиты всех форм собственности, доходов и других интересов предприятий, организаций, крестьянских хозяйств, арендаторов и граждан.

Согласно статье 2-й федерального закона от 27.11.1992г. № 4015-1 “О страховании” страхование представляет собой отношения по защите имущественных интересов физических и юридических лиц при наступлении определенных событий (страховых случаев) за счет денежных фондов, формируемых из уплачиваемых или страховых взносов (страховых премий).

Это позволяет сделать следующие выводы.

*Страхование* - это экономическое отношение, в котором участвуют как минимум две стороны (два лица, субъекта отношения).

Одна сторона (субъект) - это страховая организация (государственная, акционерная или частная), которую называют *страховщиком*. Страховщик вырабатывает условия страхования (в частности, обязуется возместить *страхователю* ущерб при страховом событии) и предлагает их своим клиентам - юридическим лицам (предприятиям, организациям, учреждениям) и физическим лицам (отдельным частным гражданам).

Если клиентов устраивают эти условия, то они подписывают *договор* страхования установленной формы и однократно или регулярно в течение согласованного периода платят страховщику *страховые премии* (платежи, взносы) в соответствии с договором.

Другая сторона (субъект) страхового экономического отношения - это юридические или физические (отдельные частные граждане) лица, называемые *страхователями*.

Согласно статье 5 закона “О страховании” страхователями признаются юридические лица и дееспособные физические лица, заключившие со страховщиками договоры страхования, либо являющиеся страхователями в силу закона.

При наступлении *страхового случая* (стихийное бедствие, падение человека с переломом и т.д.), при котором страхователю нанесен ущерб (экономический или его здоровью), страховщик в соответствии с условиями договора выплачивает страхователю компенсацию, возмещение.

Понятие договора страхования закреплено в статья 15 закона “О страховании”.

*Договор страхования* является соглашением между страхователем и страховщиком, в силу которого страховщик обязуется при страховом случае произвести страховую выплату страхователю или другому лицу, в пользу которого заключен договор страхования, а страхователь обязуется уплатить страховые взносы в установленные сроки.

Договор страхования может содержать и другие условия, определяемые по соглашению сторон, и должен отвечать общим условиям действительности сделки, предусмотренным гражданским законодательством Российской Федерации.

Страхование представляет собой отношения по защите имущественных интересов физических и юридических лиц при наступлении определенных событий (страховых случаев) за счет денежных фондов, формируемых из уплачиваемых ими страховых взносов (страховых премий).

Подытожим и определим основные термины:

* *Страховщик* - специализированная организация, производящая страхование.
* *Страхователь* - физическое или юридическое лицо, которое уплачивает страховые взносы.
* *Объекты страхования* - имущество, жизнь, здоровье, трудоспособность граждан и др.
* *Страховой полис* - документ, удостоверяющий факт страхования.
* *Страховой случай* - фактически происшедшее событие, последствием которого является выплата страхового возмещения.
* *Страховой ущерб* - ущерб, нанесенный объектам страхования в результате наступления страхового случая.
* *Страховое возмещение* - сумма ущерба, причитающаяся к выплате страхователю.
* *Страховой тариф (взнос)* - плата за страхование.
* *Страховой риск* - вероятность наступления страхового случая.
* *Страховая сумма* - величина денежных средств, на которую фактически застрахован объект страхования.
* *Срок страхования* - временной период, на который застрахованы объекты страхования.
* *Страховой портфель* - совокупная ответственность страховщика или перестраховщика по всем действующим договорам страхования или перестрахования. Фактическое число застрахованных объектов или число договоров, документально подтвержденных в делах страховщика.

Страхование осуществляется преимущественно в двух формах добровольной и обязательной.

Добровольное страхование осуществляется на основе договора между страхователем и страховщиком. Правила добровольного страхования определяются страховщиком самостоятельно в соответствии с новой редакцией закона «Об организации страхового дела в РФ». Конкретные условия страхования определяются при заключении договора страхования.

Обязательным страхованием является страхование, осуществляемое в силу закона. Виды, условия и порядок проведения обязательного страхования определяются соответствующими законами Российской Федерации.

Обязательное страхование отличается от добровольного наличием у потенциального страхователя установленной законом обязанности страховать. Иными словами, структура прав и обязанностей добровольного страхования отличается от структуры прав и обязанностей обязательного страхования.

Классификация страхования представляет собой систему деления страхования на отрасли, виды, разновидности, формы, системы страховых отношений (таблица 3.1). В основе такого деления лежат различия в объектах страхования, категориях страхования, объеме страховой ответственности и форме проведения страхования.

Отрасль страхования – это звено классификации страхования, характеризующее в широком смысле слова страхование жизни и здоровья человека, материальных ценностей, обязательств страхователей перед третьими лицами. Исходя из объектов страхования различают три отрасли страхования: личное, имущественное, страхование ответственности.

Вид страхования представляет собой часть отрасли страхования. Он характеризуется страхованием однородных имущественных интересов.

Личное страхование – это форма защиты от рисков, которые угрожают жизни человека, его трудоспособности и здоровью. Жизнь или смерть как форма существования не может быть объективно оценена. Застрахованный может лишь попытаться предотвратить те материальные трудности, с которыми столкнется в случае смерти или инвалидности.

В отличие от имущественного страхования в личном страховании страховые суммы не представляют собой стоимость нанесенных материальных убытков или ущерба, которые не могут быть объективно выражены, а определяются в соответствии с пожеланиями страхователя исходя их его материальных возможностей.

Имущественное страхование в Российской Федерации – отрасль страхования, где объектами страховых правоотношений выступает имущество в различных видах. Под имуществом понимается совокупность вещей и материальных ценностей, состоящих в собственности и оперативном управлении физического или юридического лица. В состав имущества входят деньги и ценные бумаги, а также имущественные права на получение вещей или иного имущественного удовлетворения от других лиц. Экономическое содержание имущественного страхования заключается в организации особого страхового фонда, предназначенного для возмещения ущерба его участникам, который возник в результате причинения вреда. Застрахованным может быть имущество, как являющееся собственностью страхователя (участника страхового фонда), так и находящееся в его владении, пользовании и распоряжении. Страхователями выступают не только собственники имущества, но и другие юридические и физические лица, несущие ответственность за его сохранность. Условия страхования чужого и собственного имущества могут существенно различаться, что отражено в конкретных правилах страхования. Страхование ответственности представляет собой самостоятельную сферу страховой деятельности. Объектом страхования здесь выступает ответственность страхователя по закону или в силу договорного обязательства перед третьими лицами за причинение им вреда.

Таблица 3.1. Классификация страхования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Отрасль страхования | Вид страхования | Разновидность страхования | Форма страхования | Система страховых отношений |
| Личное страхование | – страхование жизни;  – страхование от несчастных случаев и болезней;  – медицинское страхование | – страхование детей;  – страхование к бракосочетанию;  – смешанное страхование жизни и др. | – Обязательная;  – добровольная | – страхование;  – сострахование;  – двойное страхование;  – перестрахование;  – самострахование |
| Имущественное страхование | – страхование средств наземного транспорта;  – страхование средств воздушного транспорта;  – страхование средств водного транспорта;  – страхование грузов;  – страхование других видов  – имущества;  – страхование финансовых рисков | – страхование строений;  – страхование основных и оборотных фондов;  – страхование животных;  – страхование домашнего имущества;  – страхование урожая;  – страхование от потери работы, банкротства |
| Страхование ответственности | – страхование гражданской ответственности владельцев автотранспортных средств;  – страхование ответственности перевозчика;  – страхование гражданской ответственности предприятий-источников повышенной опасности;  – страхование профессиональной ответственности;  – страхование ответственности за неисполнение обязательств; | – страхование на случай причинения вреда в процессе хозяйственной и профессиональной деятельности;  – страхование убытков вследствие перерывов в производстве и др. |

Стоит отметить, что владение транспортным средством и его эксплуатация связаны с большими рисками, чем владение другим имуществом. Именно с эксплуатацией транспортного средства связаны наибольшие риски утраты и порчи собственного имущества, о также причинения вреда жизни, здоровью и имуществу третьих лиц. При этом связанные с владением автомобилем риски в меньшей степени поддаются снижению.

В большинстве стран лица, не обладающие полисом страхования автоответственности, не имеют права управлять автотранспортными средствами.

На сегодняшний день все больше и больше людей становятся владельцами собственных автомобилей. Поэтому не удивительно, что автострахование является самым популярным видом страхования в России. Именно поэтому застраховать свой автомобиль можно в большом количестве страховых компаний, представленных на рынке автострахования.

Автострахование даёт преимущество автовладельцу переложить все свои финансовые издержки при любом ДТП на свою страховую компанию.

Страхование автотранспорта является имущественным страхованием.

**3.2 ИСТОРИЯ РАЗВИТИЯ АВТОТРАНСПОРТНОГО СТРАХОВАНИЯ И ЕГО СОВРЕМЕННАЯ ОЦЕНКА**

В развитии автотранспортного страхования в нашей стране можно выделить три этапа его развития:

* + 1969 - 1985 г. - добровольное страхование средств транспорта, принадлежавшим гражданам, стало развиваться как самостоятельный вид страхования;
  + 1986 - 1990 г. - введение и развитие добровольного комбинированного страхования автомобиля, водителя и багажа (авто - комби);
  + с 1991 г. - его начало определяется введением новых Правил добровольного страхования транспортных средств и наличием условий для введения страхования гражданской ответственности участников дорожного движения.

С 1969 года добровольное страхование средств транспорта впервые стало проводиться на случай хищения, гибели или повреждений в связи с угоном транспортных средств. Были установлены льготы страхователям за безаварийную езду (лицам, страховавшим средства транспорта не менее трех лет без перерыва, предоставляется месячный льготный срок для заключения нового договора). В этот период развития добровольное страхование средств транспорта по учетным операциям стало определяться отдельно от добровольного домашнего страхования имущества.

На начало 1977 г. в целом по стране в добровольном порядке было застраховано 11% средств транспорта, имевшегося в личном пользовании, по г. Москва число договоров добровольного страхования транспортных средств достигало 86,3 тыс. и уровень охвата составил 33% т.е. в три раза превысил общесоюзный показатель.

На первом этапе развития добровольное страхование средств транспорта, принадлежащим гражданам (автокаско), завоевывало популярность в нашей стране. Правила добровольного страхования средств транспорта, принадлежащих гражданам, введенные в действие с января 1978 года расширили перечень объектов страхования и в большей степени учли потребности и интересы автовладельцев, условия эксплуатации транспортных средств, характер и причины дорожных происшествий. Договор с органами Госстраха могли заключить не только те лица, кому транспортное средство принадлежит на правах личной собственности, но и пользующиеся по доверенности или получившие его от органов социального обеспечения.

Добровольное комбинированное страхование автомобиля, водителя и багажа (авто-комби) − это вид автотранспортного страхования, положивший начало второму этапу его развития в нашей стране с 01.01.86 года. По договору авто-комби в комплексе считались застрахованными автомобили (в том числе с прицепами), багаж, находящийся в них, а также водители и страхователи автомобилей на случай смерти в результате дорожно-транспортного происшествия. С 01.01.89 года была введена ответственность страховых органов и за травму водителя и страхователя, полученную в результате дорожно-транспортного происшествия и повлекшую наступление инвалидности.

С 01.05.88 года в целях повышения эффективности и популярности автотранспортного страхования средств транспорта, заключенным сроком на один год и в страховой сумме, равной действительной стоимости транспортного средства, а так же по договора страхования авто-комби, заключенным с уплатой платежа по тарифу 2%, при определении размера причиненного ущерба за повреждение транспортного средства стоимостью новых деталей и принадлежностей не стала уменьшаться соответственно проценту износа, указанному в договоре страхования.

Введение новых правил добровольного страхования транспортных средств или наличие условий для возникновения страхования гражданской ответственности участников дорожного движения открыли дальнейшие перспективы третьего этапа развития отечественного автотранспортного страхования.

В ходе дальнейшего развития страховых отношений страхователям должна быть предоставлена широкая возможность свободно выбирать ту или иную страховую услугу с учетом своих интересов и ее качества. В условиях демонополизации страхового рынка, свободной экономической конкуренции индивидуализация предлагаемых услуг и рост их ассортимента раскроют новое качество удовлетворения страхового спроса и в автотранспортном страховании.

**3.3** **ВЗАИМОДЕЙСТВИЕ УЧАСТНИКОВ СТРАХОВЫХ ОТНОШЕНИЙ**

Средства транспорта, как и домашнее имущество, могут быть застрахованы только в добровольной форме. На страхование принимаются следующие автотранспортные средства и водные маломерные суда, подлежащие государственной регистрации в установленном порядке: автомобили - легковые, грузовые, груза - пассажирские, микроавтобусы, в том числе с прицепами (полуприцепами); мотоциклы, мотороллеры, мотоколяски, мотокарты, снегоходы, мопеды всех моделей, а также тракторы, включая мотоблоки и мини - тракторы; лодки - моторные, гребные, парусные (кроме надувных, изготовленных не под навесной мотор), катера и яхты - моторные, моторно - парусные, катамараны и тримараны.

Одновременно с транспортным средством могут быть застрахованы: водитель и пассажиры, дополнительное оборудование и принадлежности к транспортному средству, не входящих в его комплект согласно инструкции завода - изготовителя; багаж, перевозимый как на средстве транспорта, так и на его прицепе.

Страхование багажа не распространяется на антикварные и уникальные по своей сути предметы, изделия из драгоценных металлов, драгоценных и полудрагоценных и поделочных камней, предметы религиозного характера, коллекции, картины, рукописи, денежные знаки, ценные бумаги, документы и фотоснимки, а также на предметы, не принадлежащие страхователю и членам его семьи либо предназначенные для продажи в связи занятием индивидуальной трудовой или кооперативной деятельностью.

Страхователями средств транспорта являются граждане России, а также постоянно пребывающие у нас иностранцы и лица без гражданства. При этом при страховании мототранспортного средства страхователь может достичь шестнадцати летнего возраста, а других видов транспорта восемнадцати летнего возраста. Транспортное средство должно принадлежать ему на праве личной (собственности), либо взято им в аренду (напрокат), либо получено через органы социального обеспечения в установленном порядке (владелец), либо страхователь имеет от собственника (владельца) нотариально оформленную доверенность на право пользования (распоряжения) данным транспортным средством (доверенное лицо).

Свой риск (имущественный интерес) могут застраховать одновременно все указанные выше лица, эксплуатирующие данное средство транспорта.

**3.4 ОБЪЕМ СТРАХОВОЙ ОТВЕТСТВЕННОСТИ. ВАРИАНТЫ СТРАХОВАНИЯ**

Страхование средств транспорта проводится на случай наступления перечисленных ниже страховых событий (рисков) в четырех вариантах, предусматривающих по выбору страхователя полное или частичное покрытие ущерба, причиненного повреждением, уничтожением или утратой транспортного средства.

В первом варианте предусматривается полное покрытие ущерба, вызванного любым событием, за исключением повреждения шин при износе протектора выше допустимого по величине рисунка протектора, технического брака или восстановленных путем наварки протектора.

Второй вариант (предусматривается частичное покрытие) - возмещается ущерб, причиненный в результате стихийного бедствия, пожара, взрыва, дорожного (водного) - транспортного происшествия (кроме повреждения шин, если при этом транспортному средству не причинены другие повреждения), похищения или угона, попытки похищения (угона), включая кражу отдельных частей, принадлежностей и деталей транспортного средства.

Третий вариант (частичное покрытие) возмещается ущерб, причиненный в результате стихийного бедствия, дорожно (водно) - транспортного происшествия (за исключением боя стекол или повреждения шин, если при этом транспортному средству не причинены другие повреждения), похищения (угона).

Четвертый вариант (также частичное покрытие) предусматривает покрытие ущерба, причиненного пожаром, взрывом, дорожно (водно) - транспортным происшествием (кроме боя стекол или повреждения шин, если при этом транспортному средству не причинены другие повреждения.)

Независимо от варианта страхования не возмещаются убытки, произошедшие в результате военных действий, введения военного положения, гражданских волнений, радиоактивного заражения.

**3.5 МЕТОДИКА РАСЧЕТОВ СТРАХОВЫХ ТАРИФОВ. СТРУКТУРА ТАРИФНОЙ СТАВКИ**

Страховым тарифом, или тарифной ставкой, является либо денежная плата со 100 рублей страховой суммы в год, либо процентная ставка от совокупной страховой суммы на определенную дату.

С помощью тарифных ставок исчисляются страховые взносы, уплачиваемые страхователями. Страховой взнос (платеж, премия) представляет собой произведение страхового тарифа, выраженного в деньгах, на число сотен страховой суммы либо процентной тарифной ставки на совокупную страховую сумму, деленное на сто. За счет страховых платежей формируется страховой фонд, используемый для выплат страхового возмещения, а также для накладных расходов страховщика.

Страховой взнос каждого страховщика выражает его долю, его участие в формировании страхового фонда, поскольку страхование является замкнутой раскладкой ущерба между страхователями.

Основная задача, которая ставится при построении страховых тарифов, связана с определением вероятной суммы ущерба, приходящейся на каждого страхователя или на единицу страховой суммы. Если тарифная ставка достаточно отражает вероятный ущерб, то обеспечивается необходимая раскладка ущерба между страхователями.

Тарифные ставки тесно связаны с объемом страховой ответственности. Установление, расширение и ограничение объема страховой ответственности находят свое отражение в тарифных ставках. Проводя страхование, страховщик стремиться решить двоякую задачу: при минимальных тарифах, доступных для широкого круга страхователей, обеспечить достаточно значительный объем страховой ответственности. С помощью доступных тарифных ставок достигается наименьшее изъятие части доходов страхователей в виде страховых платежей в целях оказания им необходимой помощи из страхового фонда.

Тарифная ставка, лежащая в основе страхового взноса, называется брутто - ставкой. Она состоит из нетто - ставки и нагрузки к нетто - ставке. Нетто - ставка предназначена для формирования страхового фонда в его основной части, которая используется для выплат страхового возмещения. Нагрузка необходима для покрытия затрат на проведение страхования, т.е. для накладных расходов страховщика. Нагрузка составляет меньшую часть брутто - ставки (в зависимости от формы и вида страхования она колеблется то 9 до 40%).

Нагрузка к нетто - ставке включает, как правило, следующие накладные расхода страховщика: оплату труда штатных и нештатным сотрудникам страховой организации, что составляет основу всех накладных расходов; затраты на заготовку бланкового материала, пропаганду и рекламу страхового дела; административно - хозяйственные расходы, отчисления в запасные, резервные и другие фонды. В нагрузку может включаться также определенный норматив на формирование плановой прибыли от страховой деятельности.

Поскольку при страховании происходит замкнутая раскладка ущерба между страхователями, при построении нетто - ставки принято исходить из равенства (3.1):

П=В, (3.1)

где П - страховые платежи, соответствующие нетто - ставкам;

В - страховое возмещение.

При указанном равенстве, рассчитав его правую часть, получают искомую величину страховых платежей.

Если условно представить себе, что от каждого происшедшего страхового случая гибнет один застрахованный объект, то вероятность ущерба, лежащая в основе нетто - ставки, зависит, прежде всего, от вероятности наступления страховых случаев. Зная вероятное число страховых случаев за тарифный период, можно определить и степень вероятности наступления этих случаев. Она представляет собой отношение количества страховых случаев к числу застрахованных объектов. В денежном выражении числитель указанного отношения будет равен сумме страхового возмещения, а знаменатель - максимально возможному страховому возмещению, равному совокупной страховой сумме всех застрахованных объектов. Данное отношение есть показатель убыточности страховой суммы. Поскольку числитель этого показателя меньше знаменателя, его значение всегда меньше единицы. Для расчета нагрузки применяется формула (3.2):

Нагрузка = B – N, (3.2)

где B - брутто - ставка,

N - нетто - ставка.

В свою очередь, брутто - ставку можно рассчитать по формуле (3.3):

B = (N/(100 - H(100%))\*100, (3.3)

где Н (%) - удельный вес нагрузки в брутто - ставке, определенный на основе расчета фактических накладных расходов страховщика за последние 1 - 2 года.

Правилами страхователю автомобиля предоставляется возможность заключить договор при условии:

* 1. собственного участия в возмещении ущерба (с франшизой). Размер франшизы выбирает страхователь. При этом страховой платеж по договору соответственно уменьшается, а ущерб, причиненный автомобилю, дополнительному оборудованию и предмета багажа менее установленной суммы франшизы, не возмещается. В случае, если ущерб будет больше установленной суммы франшизы, он подлежит возмещению в полном объеме;
  2. возмещения ущерба без учета скидки на износ подлежащих замене запасных частей, деталей и принадлежностей в случае их повреждения (с уплатой дополнительного платежа);
  3. возмещения стоимости потери товарного вида в результате страховых событий, на случай наступления которых заключен договор (также с уплатой дополнительного платежа), но при условии, что автомобиль принят на страхование в размере действительной стоимости.

Дополнительный платеж при страховании автомобиля без учета скидки на износ исчисляется в следующем размере от суммы платежа:

* + 5% - если срок его эксплуатации 5 лет;
  + 10% - если срок эксплуатации свыше 5 лет и до 8лет;
  + 20% - свыше 8 лет и до 12 лет;
  + 30% - свыше 12 лет.

Соответственно при страховании автомобиля с условием возмещения стоимости потери товарного вида дополнительный платеж уплачивается в следующем размере от исчисленной суммы:

* + 20% - если срок эксплуатации автомобиля до 5лет;
  + 15% - если свыше 5 лет до 8 лет;
  + 10% - если 8 и до 12 лет;
  + 5% - если свыше 12 лет.

Договор страхования заключается сроком на 1 год, а также на срок одного до 11 полных месяцев. Действие договора начинается, со следующего дня после уплаты страхового платежа наличными деньгами, при безналичном расчете - со дня выдачи заработанной платы, установленной в организации, где страхователь работает, а при заключении договора в магазине - со дня получения страхователем транспортного средства.

При неуплате страхового взноса в два срока действие договора прекращается через 4 месяца после вступления его в силу, если к тому времени независимо от причины не внесена вторая половина платежа.

**3.6 СУЩНОСТЬ И ВИДЫ СТРАХОВОГО ПОРТФЕЛЯ**

Под управлением страховым портфелем понимается применение к совокупности различных видов страховых услуг определенных методов и технологических возможностей, которые позволяют сохранить первоначальный уставный капитал, достичь максимального уровня дохода, обеспечить финансовую устойчивость страхового портфеля, обеспечить минимизацию риска.

Для этого необходим пересмотр портфеля не реже одного раза в год, а при стабильном портфеле страховщика, работающего на развитом, сложившемся рынке — в три-пять лет. Помимо этого, необходима текущая корректировка структуры портфеля на основе мониторинга факторов, которые могут вызвать изменение в составных частях портфеля.

Совокупность применяемых при формировании портфеля методов и технических возможностей и есть способ управления (активный и пассивный). Это один из наиболее дорогостоящих, трудоемких элементов управления, представляющий собой непрерывный, детальный анализ: рынка страхования и тенденций его развития, сегментации страхового рынка, финансово экономических показателей страховых компаний, емкости страхового рынка и степени освоения. Это основа активного и пассивного способов управления.

Активная модель управления предлагает тщательное отслеживание и немедленное использование финансовых инструментов, отвечающих целям формирования портфеля, а также быстрое изменение состава страховых услуг, входящих в портфель. Отечественный рынок страхования характеризуется динамичностью процессов, высоким уровнем риска. Считается, что его состояние адекватно активной модели мониторинга, которая делает управление портфелем эффективным.

Андеррайтер, занимающийся активным управлением и формированием страхового портфеля, должен оценить приемлемое сочетание риска и дохода портфеля, определить удельный вес договоров в страховом портфеле с различными уровнями риска и дохода по видам страхования. При этом важно не допустить снижения стоимости страхового портфеля, а для этого необходимо сопоставлять доходность и риск «нового» портфеля с учетом вновь реализованного пакета страховых услуг и аналогичными характеристиками имеющегося «старого» портфеля.

Высокая конъюнктура страхового рынка диктует необходимость сократить срок существования уже сформированного страхового портфеля. Чем протяженнее срок, в течение которого не пересматривается страховой портфель, тем больше стоимость портфеля подвержена колебаниям вследствие изменения макроэкономических показателей. Впрочем, надо заметить, что портфель договоров по страхованию жизни — наиболее долгосрочный и устойчивый.

Сущность и необходимость финансовой устойчивости страхового портфеля проявляется в функциях управления им. Это функции: отбора страховых услуг; диверсификации страхового портфеля; расчетная (себестоимость страхового портфеля); ревизионная; оптимизации «нового портфеля».

Все эти функции сводятся к созданию эффективного страхового портфеля, достижению оптимального соотношения между доходом и степенью риска, т.е. всего того, что в конечном итоге будет служить стабилизирующим фактором финансовой устойчивости страхового портфеля.

Функция отбора страховых услуг предполагает право страховщика избрать такой тип портфеля страхования, который обеспечивал бы определенное качество и ассортимент страховых услуг, отвечающих потребностям страхователей.

С рассматриваемой функцией связан такой показатель, как величина страхового портфеля, характеризующий как число застрахованных объектов, так и объем страховой ответственности.

Функция диверсификации сводится к структурному формированию портфеля. Структура определяется соотношением между формами (обязательной и добровольной, индивидуальной и групповой) и системой видов страховых продуктов.

Расчетная функция страхового портфеля выражается многими составляющими: себестоимостью страхового портфеля, равновесием, при котором приток новых договоров компенсирует заканчивающиеся, числом договоров и суммой взносов по ним, страховой суммой, сроком страхования и вероятностью ущерба.

Ревизионная функция — необходимый атрибут контрольной функции в страховом деле вообще, и в страховом портфеле в частности. Эта функция включает в себя пересмотр существующего портфеля в целях анализа эффективности действующего портфеля с точки зрения доходности и риска, оптимизацию и выравнивание доходности, выработку рекомендаций и предложений.

Реализацией функции формирования завершается формирование «нового» портфеля страхования, расчет «нового» портфеля страхования, его доходности и риска, с учетом ротаций предыдущего, «старого» портфеля.

Главная цель формирования страхового портфеля состоит в достижении оптимального сочетания между риском и доходом страховщика. Методом снижения риска потерь служит диверсификация, т.е. формирование структуры страхового портфеля по определенному типу.

В зависимости от степени риска портфели делятся на: специализированные, классические и комбинированные.

Понятно, что каждому типу страхового портфеля будет соответствовать и определенный уровень дохода: агрессивному типу (специализированному) — высокий уровень; умеренному (классическому) — устойчивый, но низкий доход; комбинированному типу — диверсификационный доход.

При рассмотрении структуры страхового портфеля необходимо учитывать такие факторы, как отраслевая особенность, освобождение от налогов (обязательное страхование), формы страхования, краткосрочные и долгосрочные договоры, тарифная политика, система перестрахования, страховые фонды и страховые резервы.

Вложение, связанное с формированием портфеля, представляет собой совокупность информационных ценностей, служащих инструментом для достижения конкретной цели вкладчика — достижения оптимального сочетания между доходом и риском.

Риск в страховании это, во-первых, конкретное явление, при наступлении которого производятся выплаты из страхового фонда; во-вторых, риск связан с конкретным застрахованным объектом и поэтому имеет конкретное проявление. По отношению к объекту проявляются и изучаются факторы риска. Анализ полученной информации позволяет добиться существенного снижения или предотвращения риска, в-третьих, риск сопряжен с вероятностью гибели или повреждения данного объекта, принятого на страхование.

Выравнивание риска, распределение риска, разделение риска и составляют технические приемы страховщиков, с помощью которых формируются страховые портфели. Выбор конкретных технических приемов зависит от размера риска. Правильная оценка его связана с имеющимися ресурсами страхового фонда и возмещением материального ущерба страхователю.

Наиболее трудной задачей страховщика является поддержание соответствующей тарифной политики, прогнозируемым тенденциям в развитии риска.

Выделяют следующие виды риска: риски, которые возможно застраховать; риски, которые невозможно застраховать; благоприятные и неблагоприятные риски; технический риск страховщика.

Страховой риск, по определению профессора Л.И.Рейтмана, трактуется как «термин, имеющий четыре смысловых значения.

1. Вероятность нанесения ущерба от страхового случая, исчисленная математически, эта вероятность является основой для построения страховых тарифов.

2. Конкретный страховой случай, т.е. определенная опасность, от которой проводится страхование. В таком понимании перечень страховых рисков составляет объект страховой ответственности.

3. Часть стоимости имущества, не охваченная страхованием и оставляемая тем самым на риске страхователя.

4. Конкретные объекты страхования по их страховой оценке и степени вероятности нанесения ущерба.

В этом понимании различают крупные, средние и мелкие страховые риски, в зависимости от величины их страховой оценки, а также более опасные и менее опасные риски по степени вероятности их гибели или повреждения.

На сегодняшний день существует много определений термина «страховой портфель». Предоставляется наиболее интересным определение Л.И.Рейтмана, который писал, что «страховой портфель — фактическое количество застрахованных объектов или действующих договоров страхования на данной территории или на предприятии».

Исходя из этого определения, страховой портфель − стоимостной баланс страховых рисков и страхового покрытия рисков по определенной совокупности договоров (объектов) страхования.

При создании страхового портфеля страховщик или андеррайтер должен принять решения по следующим вопросам.

1. Какой тип портфеля по степени риска выбрать.

а) высокая степень риска — агрессивный тип;

б) невысокая степень риска — умеренный тип;

в) низкая степень риска — консервативный тип.

2. Оценить приемлемое сочетание риска и доходы страхового портфеля, определить удельный вес портфеля страхования с различными сочетаниями риска и дохода.

а) классический (набор традиционных страховых услуг);

б) специализированный (набор «новых» или узконаправленных страховых продуктов);

в) комбинированный (сочетание традиционного и специализированного набора страховых услуг).

Классический страховой портфель состоит из набора традиционных страховых услуг, которые длительное время составляют основу страхования в России. Он, как правило, содержит договоры обязательного и добровольного страхования по личному и имущественному страхованию, при этом перечень оказываемых услуг страхования традиционен.

В области страхования гражданской ответственности — страхование гражданской ответственности владельцев автотранспортных средств, страхование гражданской ответственности предприятий — источников повышенной опасности.

Как правило, традиционный классический портфель наиболее устойчив в финансовом отношении и по степени риска и относится к консервативному типу.

Доходность по классическому виду страхового портфеля — устойчивая, невысокая при умеренном риске. Необходимо также отметить, что для занятия традиционным видом страхования необходимо иметь большой размер страхового фонда и собственных средств у страховщика.

Специализированный тип страхового портфеля более динамичен, узконаправлен, имеет высокую степень риска и относится к агрессивному типу портфеля, имеет высокий доход. Это: космическое страхование, страхование политических рисков, страхование авиаперевозок, морское страхование, обязательное и добровольное медицинское страхование, страхование специфических рисков, страхование профессиональной ответственности, страхование инвестиций, страхование биржевых и валютных рисков.

Например, страхование политических рисков — это очень специфический вид страховой услуги. К ним относят события непреодолимой силы, которые имеют политический характер и исходят от органов власти и управления.

Перечень политических рисков обычно относят в договоре страхования к «форс-мажорной» оговорке, согласно которой нанесение ущерба имущественным интересам страхователя при наступлении перечисленных форс-мажорных событий не влечет за собой возникновение обязанностей страховщика компенсировать нанесенный ущерб. В договоре могут быть оговорены такие риски, как: конфискация, национализация собственности, военные действия, гражданские беспорядки, повлекшие за собой причинения ущерба имущественным интересам инвестора. Страхование имущественных интересов инвестора от политических рисков может осуществляться государственными агентствами, международными организациями или крупными частными страховыми компаниями.

Комбинированный тип страхового портфеля сочетает в себе элементы классического и специализированного портфеля страховых услуг. При выборе этого вида портфеля вероятность риска невысокая — перед нами умеренный тип страхового портфеля.

Тип этого портфеля выбирают, как правило, страховщики, только что появившиеся на рынке страхования и до конца не определившие своего направления.

Тип портфеля в зависимости от степени риска — это его характеристика, основанная на соотношении дохода и риска. Выделены следующие типы портфеля: агрессивный, консервативный, умеренно-агрессивный. Виды портфеля — сбалансированный (диверсификационный), классический (традиционный) и специализированный (не сбалансированный).

Типы страхового портфеля по степени доходности и устойчивости — высокоустойчивый, но низкодоходный; диверсификационный, но среднедоходный; рискованный, но высокодоходный и низкоустойчивый.

1. **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**
   1. **КЛАСТЕРИЗАЦИЯ СТРУКТУРЫ ПОРТФЕЛЯ АВТОТРАНСПОРТНОГО СТРАХОВАНИЯ ФИЗИЧЕСКИХ ЛИЦ**

Для прогнозирования портфеля автострахования существует несколько методов и методик. Наибольшей популярностью из них пользуется методика андеррайтинга.

Начальная стадия методики андеррайтинга для страхования автотранспортных средств физических лиц заключается в изучении структуры страхового портфеля, его убыточности и факторов, влияющих на эффективность страхования. Проведенный анализ позволит прогнозировать показатели страхового портфеля в случае изменения его структуры. С этой целью определяются кластеры по совокупности значимых взаимонезависимых факторов риска в соответствии с разбросом цен на рынке автотранспортных средств и статистикой страховой компании.

Многочисленные исследования рисков по автотранспортным средствам позволили выделить 9 факторов, отражающих состояние и условия эксплуатации застрахованных объектов:

X1 – стоимость автомобиля;

X2 – средний убыток по объекту;

X3 – срок эксплуатации;

X4 – износ;

X5 – мощность;

X6 – масса автомобиля;

X7 – возраст водителя;

X8 – стаж водителя;

X9 – риск ДТП;

Таким образом задача анализа данных сводится к задаче кластеризации.

Кластеризация структуры портфеля автотранспортного страхования позволяет на начальных стадиях процесса андеррайтинга выявить неэффективность страхового портфеля, ошибки в тарифном руководстве.

* 1. **АЛГОРИТМ K-СРЕДНИХ**

Первый из используемых алгоритмов - алгоритм k-средних, один из самых популярных алгоритмов кластеризации.

Суть метода *k*-средних состоит в исследовании заранее определенного количества классов *k*, на которые необходимо разбить имеющиеся наблюдения, и начальных центров этих классов. Для каждого следующего наблюдения рассчитываются расстояния до центров кластеров, и данное наблюдение относится к тому кластеру, расстояние до которого было минимальным. После этого для данного кластера (в котором увеличилось количество наблюдений) рассчитывается новый центр тяжести (как среднее по каждому показателю) по всем включенным в кластер наблюдениям. В качестве меры расстояния чаще всего используют суммарное квадратичное отклонение точек кластеров от центров этих кластеров, которое алгоритм и стремится минимизировать. Алгоритм завершается, когда на какой-то итерации не происходит изменения центра масс кластеров.Это происходит за конечное число итераций, так как количетсов возможных разбиений конечно, а на каждом шаге суммарное квадратичное отклонение уменьшается, поэтому зацикливание не возможно. Но на практике количество итераций ограничивают определенным числом, для большего ускорения работы алгоритма.

Достоинствами алгоритма являются быстрота и простота реализации.

К недостаткам алгоритма можно отнести то, что необходимо определить заранее количество кластеров разбиения. Данный недостаток устраняется одним из двух способов:

* + 1. предварительный анализ исходных данных.

Исходные данные стандартизируются так, чтобы каждая переменная имела среднее 0 и стандартное отклонение 1, и выясняется, формируют ли данные «естественные» кластеры, которые могут быть осмыслены.

В качестве правила объединения выбирается метод Уорда, в качестве меры близости – евклидово расстояние.

Метод Уорда предполагает, что каждый объект – это отдельный кластер. Рассчитывается расстояние между объектами и в кластер объединяются наиболее «близкие» объекты. Метод Уорда приводит к образованию кластеров приблизительно равных размеров с минимальной внутрикластерной дисперсией.

Мера близости, определяемая евклидовым расстоянием, является геометрическим расстоянием в *n*-мерном пространстве и вычисляется по формуле (4.1):

(4.1)

где *d(x,y)* – евклидово расстояние между *i*-тым и *j*-тым объектами;

*xik* – численное значение *k*-той переменной для *i*-того объекта;

*xjk* – численное значение *k*-той переменой для *j*-того объекта;

*n*– количество переменных, которыми описываются объекты.

Наиболее важным результатом, получаемым в результате, является иерархическое дерево – дендрограмма. Каждый узел дендрограммы представляет объединение двух или более кластеров, положение узлов на вертикальной оси определяет расстояние, на котором были объединены соответствующие кластеры. Выбор количества кластеров разбиения для алгоритма k-средних осуществляется на основе анализа дендрограммы.

* + 1. последовательный запуск алгоритма и выбор наилучшего разбиения.

Данный способ на практике применяется чаще всего. Задаются начальное и конечное число кластеров разбиения. Далее осуществляется последовательный запуск алгоритма с разбиением на кластеры, в соответствии с заданными значениями. Наилучшее разбиение выбирается по определенному правилу, например, по минимальному значению суммарного квадратичного отклонения точек кластеров от центров этих кластеров среди всех запусков алгоритма. В данной работе используется именно этот способ.

Таким образом общий алгоритм можно описать следующим образом:

1. Задается начальное и конечное число кластеров разбиения, kmin, kmax. Запуск общего алгоритма.
2. Случайным образом выбираются С точек. которые считаются центрами кластеров.
3. Запуск алгоритма k-средних. Осуществляется итеративное распределение объектов по кластерам. Объекты распределяются по кластерам путем подсчета расстояний от объекта до центра кластеров и выбора наименьшего:
4. Вычисление новых центров кластеров и суммарного квадратичного отклонения.
5. Если центры кластеров не изменены и суммарное квадратичное отклонение минимально среди всех итераций алгоритма k-средних, то работа алгоритма завершается, в противном случае повторяются шаги 3-4.
6. Повторяются шаги 2-5 в соответствии с заданными kmin и kmax.
7. За искомое количество кластеров разбиения исходных данных выбирается то, на шаге которого суммарное квадратичное отклонение в результате работы алгоритма k-средних было минимальным. Результаты разбиения на этом шаге принимаются за искомые.

Блок-схема алгоритма (рисунок 4.1):

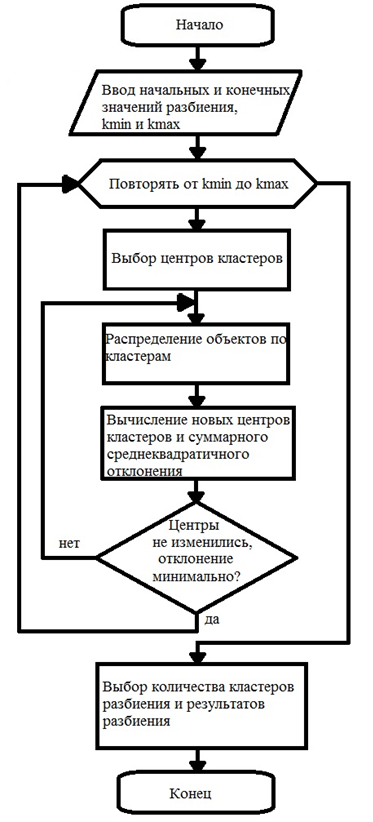


Рисунок 4.1 – Блок-схема алгоритма k-средних

* 1. **АЛГОРИТМ CLOPE**

В алгоритме CLOPE вводится новый термин транзакция, в определении которого отражена специфика обрабатываемых данных. Под транзакциями понимается некоторый произвольный набор объектов, будь это список ключевых слов статьи, товары, купленные в супермаркете, множество симптомов пациента, характерные фрагменты изображения и так далее. Но, не смотря на ориентированность алгоритма CLOPE на транзакции, его можно применять к любым данным.

В основе алгоритма кластеризации CLOPE лежит идея максимизации глобальной функции стоимости, которая повышает близость транзакций в кластерах при помощи увеличения параметра кластерной гистограммы. Рассмотрим пример из 5 транзакций: . Сравним между собой следующие два разбиения на кластеры:

****

Для первого и второго вариантов разбиения в каждом кластере рассчитаем количество вхождений в него каждого элемента транзакции, а затем вычислим высоту  и ширину  кластера. Например, кластер  имеет вхождения , где , а . Для облегчения понимания на рисунке 4.2 эти результаты показаны геометрически в виде гистограмм.

**а)б)**

Рисунок 4.2 – Гистограммы двух разбиений

Оценим качество двух разбиений, проанализировав их высоту  и ширину . Кластеры  и  имеют одинаковые гистограммы, следовательно, равноценны. Гистограмма для кластера  содержит 4 различных элемента и имеет площадь 8 блоков (), а кластер  – 5 различных элементов с такой же площадью (). Очевидно, что разбиение *а)* лучше, поскольку обеспечивает большее наложение транзакций друг на друга.

Рассмотрим алгоритм CLOPE в более формальном описании. Пусть имеется база транзакций , состоящая из множества транзакций . Каждая транзакция есть набор объектов . Множество кластеров  есть разбиение множества , такое, что  и  и  . Каждый элемент  называется кластеров,  – количество транзакций, количество объектов в базе транзакций и число кластеров соответственно.

Каждый кластер  имеет следующие характеристики:  – множество уникальных объектов,  – количество вхождений (частота) объекта  в кластер .

**;** (4.2)

**;** (4.3)

**.** (4.4)

****

Рисунок 4.3 – Иллюстрация гистограммы кластера

Гистограммой кластера  называется графическое изображение его расчетных характеристик: по оси  откладываются объекты кластера в порядке убывания величины , а сама величина  – по оси  (рисунок 3). На рисунке 3 , равное 8, соответствует площади прямоугольника, ограниченного осями координат и пунктирной линией. Формула для вычисления глобального критерия – функции стоимости :

, (4.5)

где  – количество объектов в -ом кластере,  – коэффициент отталкивания, вещественное число большее 1, с помощью которого регулируется уровень сходства транзакций внутри кластера, и, как следствие, финальное количество кластеров.

Таким образом можно дать следующее описание алгоритма:

1. Задается коэффициент отталкивания.
2. Последовательно считываются данные и помещаются в новый либо существующий кластер: рассчитываются характеристики очередного кластера, если функция стоимости выросла, транзакцию оставляется в нем, если нет – добавляется в следующий кластер и снова рассчитываются характеристики.
3. Последовательное перемещение каждой транзакции из текущего кластера в новый либо существующий кластер, если это увеличивает значение функции стоимости. Если перемещений не было, то кластеризация считается законченной, в противном случае необходимо выполнить этап 3 заново с самого начала.
4. Удаление пустых кластеров и применение результатов.

Блок-схема алгоритма (рисунок 4.4):

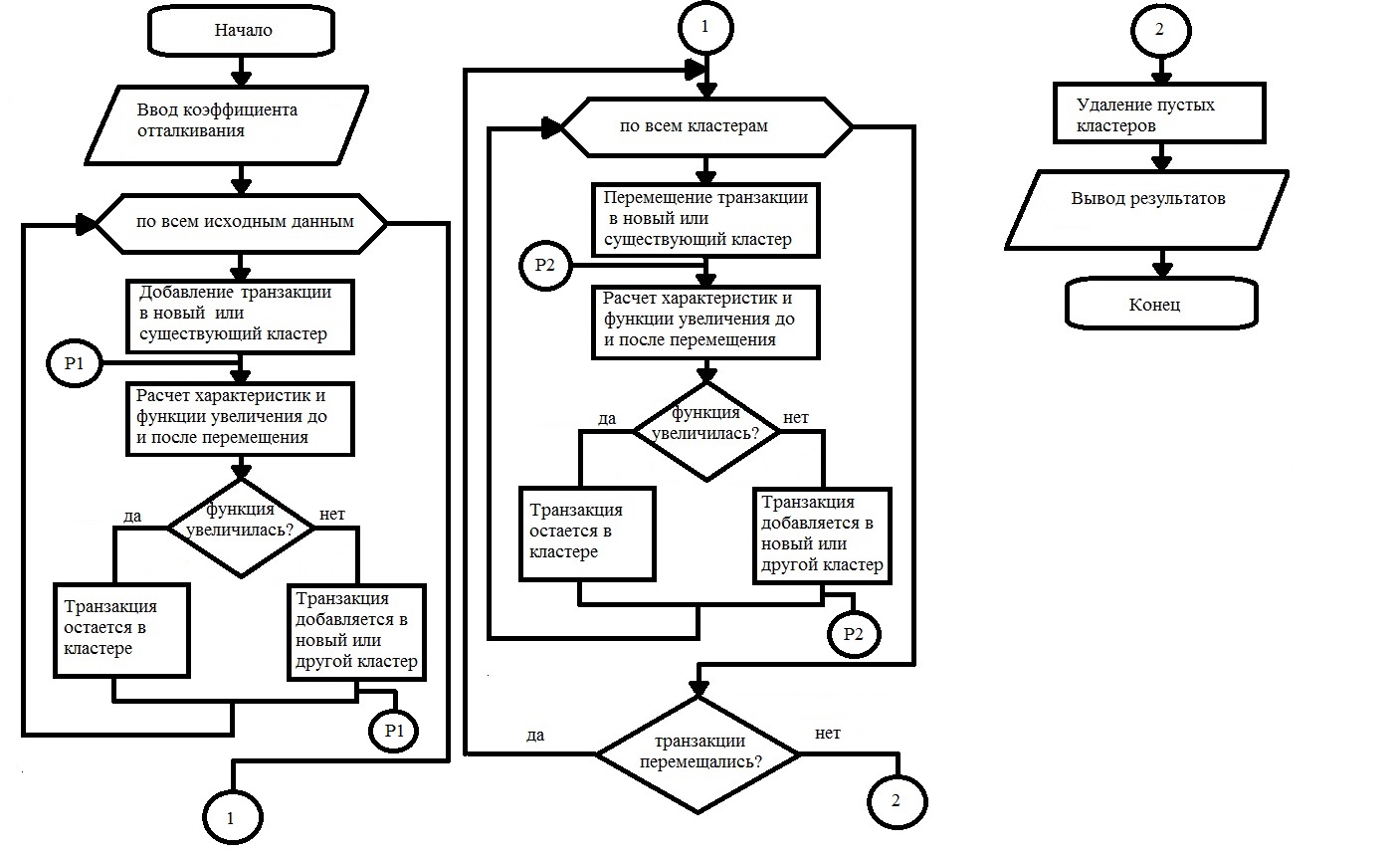


Рисунок 4.4 – Блок-схема алгоритма CLOPE

* 1. **ОПИСАНИЕ ЯЗЫКА И СРЕДСТВ**

**ПРОГРАММИРОВАНИЯ**

* + 1. **ЯЗЫК ПРОГРАММИРОВАНИЯ C#**

C# – объектно-ориентированный язык программирования, применяемый для решения широкого круга задач.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.

Переняв многое от своих предшественников – языков C++, Java, Delphi, Модула и Smalltalk – С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем: так, C# не поддерживает множественное наследование классов (в отличие от C++).

Язык С# и связанную с ним среду .NET Framework можно без преувеличения назвать самой значительной из предлагаемых в настоящее время технологий для разработчиков. Среда .NET является такой средой, которая была создана для того, чтобы в ней можно было разрабатывать практически любое приложение для запуска в Windows, а С# является языком программирования, который был специально создан для использования в .NET Framework [18].

С# является одним из языков Microsoft Visual Studio – интегрированной среды разработки программного обеспечения. Данный продукт позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологий Windows Forms, Windows Presentation Foundation, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework и Microsoft Silverlight [21].

**4.4.2 WINDOWS PRESENTATION FOUNDATION**

Windows Presentation Foundation (WPF) – это система следующего после Windows Forms поколения для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем. С помощью WPF можно создавать широкий спектр как автономных, так и размещенных в браузере приложений.

В основе WPF лежит векторная система отрисовки, не зависящая от разрешения и созданная с расчетом на возможности современного графического оборудования. WPF расширяет базовую систему полным набором функций разработки приложений, в том числе Язык XAML (Extensible Application Markup Language), элементами управления, привязкой данных, макетом, двухмерный- и трехмерный-графикой, анимацией, стилями, шаблонами, документами, мультимедиа, текстом и оформлением. WPF входит в состав Microsoft .NET Framework и позволяет создавать приложения, включающие другие элементы библиотеки классов .NET Framework – фактически использовать стандартные элементы управления в WPF так же легко, как, например, и в Windows Forms. Более того, WPF расширяет средства, адресованные именно бизнес-разработчикам, включая значительно усовершенствованную модель привязки данных, набор классов для печати содержимого и управления очередями печати, а также средства работы с документами для отображения огромных объемов форматированного текста. Доступна даже модель для построения приложений на основе страниц, которые гладко работают в любом интернет-браузере [13].

Делая вывод, можно сказать, что WPF объединяет лучшее из мира разработки под Windows с новейшими технологиями для построения современных, графически развитых пользовательских интерфейсов.

1. **ПРОЕКТНАЯ ЧАСТЬ**

Цель данного дипломного проекта заключается в повышении точности прогнозирования портфеля автострахования посредством разработки программного обеспечения, основанного на применении методов анализа данных.

На базе разработанного программного обеспечения в дальнейшем существует возможность построения информационной системы (ИС), предполагающей интеграцию базы данных, СУБД и приложения.

На основании этого в процессе разработки программного обеспечения проектирование структуры программных компонентов в данном дипломном проекте осуществляется с использованием инструментальных средств проектирования − CASE-технологий.

Под термином CASE-технологии понимаются программные средства, поддерживающие процессы создания и сопровождения ИС, включая анализ и формулировку требований, проектирование прикладных приложений и баз данных, тестирование, документирование, генерацию кода, обеспечение качества, конфигурационное управление и управление проектом, а также другие процессы.

Большинство существующих CASE-средств основано на методологиях структурного или объектно-ориентированного анализа и проектирования, использующих спецификации в виде диаграмм или текстов для описания внешних требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств.

Для успешной реализации проекта объект проектирования должен быть прежде всего адекватно описан, должны быть построены полные и непротиворечивые функциональные и информационные модели предметной области.

Накопленный к настоящему времени опыт проектирования показывает, что это логически сложная, трудоемкая и длительная по времени работа, требующая высокой квалификации участвующих в ней специалистов.

Разумный подход к использованию CASE-средств может преодолеть все перечисленные трудности.

Успешное внедрение CASE-средств должно обеспечить такие выгоды как:

* положительное воздействие на некоторые или все из перечисленных факторов: производительность, качество продукции, соблюдение стандартов, документирование [7].
* высокий уровень технологической поддержки процессов разработки и сопровождения программного обеспечения;

**5.1 ДИАГРАММА ПРЕЦЕДЕНТОВ ИСПОЛЬЗОВАНИЯ**

Диаграмма прецедентов использования описывает функциональное назначение программного обеспечения - то, что программа будет делать в процессе своего функционирования, с точки зрения пользователя.

Проектируемое программное обеспечение представляется в виде множества сущностей или актеров, взаимодействующих с ним с помощью так называемых вариантов использования.

При этом актером или действующим лицом называется любая сущность, взаимодействующая с программным обеспечением.

В качестве актера проектируемого программного обеспечения выступает пользователь и программист.

Каждый из них взаимодействует с программным обеспечением.

|  |
| --- |
| C:\Users\Владимир\Documents\!Работа\7Башева-CLOPE\UML\П.jpg  Рисунок 5.1 – Диаграмма прецедентов использования |

Поэтому, прежде чем приступить к анализу вариантов использования, важно описать роли, которые играют основные действующие лица (далее приведено описание основных ролей (в бессистемном стиле) для каждого актера).

1. Для пользователя:

Пользователь запускает программу, производит настройку алгоритмов, загружает исходные данные для прогнозирования, выбирает и запускает один из алгоритмов анализа. Результаты можно представить в виде отчета, который можно вывести на экран, а затем на печать.

Пользователь самостоятельно подготавливает необходимые исходные данные в требуемом для программы виде.

2. Для программиста:

Программист осуществляет установку программного обеспечения и сопровождает его в процессе эксплуатации.

Диаграмма вариантов использования, между которыми установлены отношения включения и расширения, изображена на рисунке 5.1.

Следует отметить, что приведенная ниже диаграмма вариантов использования не является окончательной. Она может быть детализирована далее, с целью более глубокого уточнения предъявляемых к программному обеспечению требований.

* 1. **ДИАГРАММА КЛАССОВ**

Диаграмма классов служит для представления статической структуры модели программного обеспечения в терминологии классов объектно-ориентированного программирования. Класс – абстрактное описание множества однородных объектов, имеющих одинаковые атрибуты, операции и отношения с объектами других классов [7].

На основе диаграммы прецедентов использования и анализе предметной области были спроектированы классы, изображенные на диаграмме классов (рисунок 5.2).

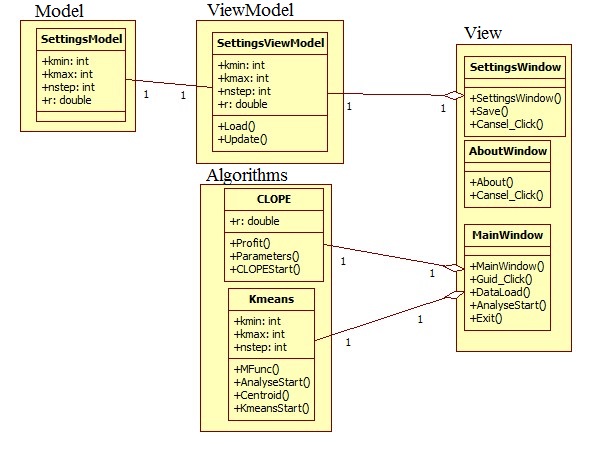


Рисунок 5.2 – Диаграмма классов

Атрибуты и операции для каждого из классов выбирались, исходя из того, чтобы они адекватно и полно описывали свойства и основные варианты поведения объектов предметной области.

* + 1. **ОПИСАНИЕ КЛАССОВ**

**Класс SettingsModel**

1. SettingsModel-класс модели для окна настроек
2. Атрибуты:

* *kmin* (задает начальную величину разбиения для алгоритма k-средних)
* *kmax* (задает конечную величину разбиения для алгоритма k-средних)
* *nstep* (задает максимальное число итераций для алгоритма k-средних)
* *r* (задает коэффициент оттталкивания для алгоритма CLOPE)

**Класс SettingsViewModel**

1. SettingsViewModel-класс модели-представления для окна настроек
2. Атрибуты:

* *kmin* (задает начальную величину разбиения для алгоритма k-средних)
* *kmax* (задает конечную величину разбиения для алгоритма k-средних)
* *nstep* (задает максимальное число итераций для алгоритма k-средних)
* *r* (задает коэффициент оттталкивания для алгоритма CLOPE)

1. Операции:

* *Load* (реализует загрузку настроек из файла ресурсов)
* *Update* (реализует обновление настроек в файле ресурсов)

**Класс CLOPE**

* + 1. CLOPE-класс алгоритма CLOPE
    2. Атрибуты:
* *r* (задает коэффициент оттталкивания для алгоритма CLOPE)
  + 1. Операции:
* *Profit* (реализует вычисление функции увеличения)
* *Parameters* (реализует вычисление параметров кластера)
* CLOPEStart (реализует запуск основного алгоритма)

**Класс Kmeans**

1. Kmeans-класс алгоритма k-средних
2. Атрибуты:

* *kmin* (задает начальную величину разбиения для алгоритма k-средних)
* *kmax* (задает конечную величину разбиения для алгоритма k-средних)
* *nstep* (задает максимальное число итераций для алгоритма k-средних)

1. Операции:

* *MFunc* (реализует вычисление среднеквадратичного отклонения)
* *AnalyseStart* (реализует запуск алгоритма по заданным значениям разбиения)
* *Centroid* (реализует вычисление центроидов кластеров)
* *KMeansStart* (реализует запуск алгоритма k-средних)

**Класс SettingsWindow**

1. Класс окна настроек программы
2. Операции:

* *SettingsWindow* (конструктор класса)
* *Save* (реализует сохранение настроек)
* *Cansel\_Click* (реализует закрытие окна настроек)

**Класс MainWindow**

1. Класс главного окна программы
2. Операции:

* *MainWindow* (конструктор класса)
* *Guide\_Clik* (реализует открытие руководства пользователя)
* *Data\_Load* (реализует загрузку исходных данных)
* *AnalyseStart* (реализует запуск анализа данных)
* *Exit* (реализует завершение работы программы)

**Класс AboutWindow**

1. Класс окна «О программе»
2. Операции:

* *About* (конструктор класса)
* *Cansel\_Click* (реализует закрытие окна о программе)
  + 1. **ОПИСАНИЕ ОТНОШЕНИЙ МЕЖДУ КЛАССАМИ**

**Ассоциация SettingsModel – SettingsViewModel**

1) Имя ассоциации: *Соответствует*

2) Направление чтения ассоциации: SettingsViewModel – *Соответствует* – SettingsModel*.*

3) Мощность ассоциации: 1.. 1 (классу SettingsViewModel соответствует один экземпляр класса SettingsModel)

**Агрегация SettingsViewModel – SettingsWindow**

1) Имя агрегации: *Включает*

2) Направление чтения агрегации: SettingsWindow – *Включает* – SettingsViewModel*.*

3) Мощность ассоциации: 1.. 1 (класс SettingsWindow включает один экземпляр класса SettingsViewModel)

**Агрегация CLOPE – MainWindow**

1) Имя агрегации: *Включает*

2) Направление чтения агрегации: MainWindow – *Включает* – CLOPE*.*

3) Мощность ассоциации: 1.. 1 (класс MainWindow включает один экземпляр класса CLOPE)

**Агрегация Kmeans – MainWindow**

1) Имя ассоциации: *Включает*

2) Направление чтения ассоциации: MainWindow – *Включает* – Kmeans*.*

3) Мощность ассоциации: 1.. 1 (класс MainWindow включает один экземпляр класса Kmeans)

* 1. **ДИАГРАММА ВЗАИМОДЕЙСТВИЯ ОБЪЕКТОВ**

Диаграммы взаимодействияописывают взаимодействие групп объектов в различных условиях их поведения. UML определяет диаграммы взаимодействия нескольких типов, из которых наиболее употребительными являются диаграммы последовательности.

Обычно диаграмма последовательности описывает один сценарий.

На диаграмме показаны экземпляры объектов и сообщения, которыми обмениваются объекты в рамках одного прецедента (use case).

На рисунке 5.3 приведена диаграмма последовательности.

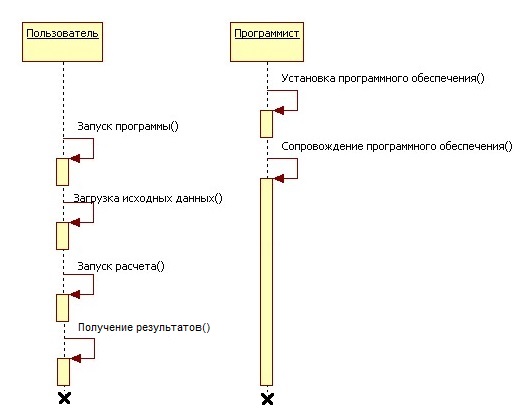


Рисунок 5.3 – Диаграмма последовательности

**5.4 ДИАГРАММА ДЕЯТЕЛЬНОСТИ**

Диаграммы взаимодействий не отражают детально порядок выполнения операций в части разветвлений, циклических повторений, параллельности или произвольности действий. Данные недостатки исправляет диаграмма деятельностей.

Под деятельностью понимается некоторая работа, которую можно декомпозировать на совокупность действий.

Каждое состояние на диаграмме деятельности соответствует выполнению некой операции, а переход в следующее состояние происходит только после завершения выполнения этой операции.

Диаграмма деятельностей может отражать взаимодействие объектов из нескольких прецедентов использования, в частности, реализующих отдельно стандартные и альтернативные пути обработки объектов [7].

Диаграмма деятельности представлена на рисунке 5.4.

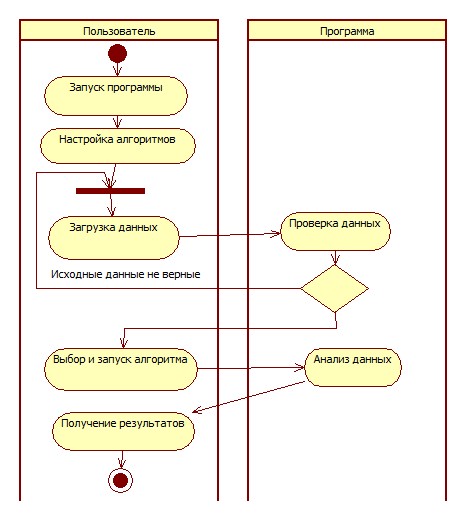


Рисунок 5.4 – Диаграмма деятельности

Спецификация для указанной диаграммы представлена в таблице 5.1.

Таблица 5.1 – Спецификация диаграммы деятельности

|  |  |
| --- | --- |
| Название: | Получение прогноза |
| Действующие лица: | Пользователь, Программа |
| Краткое описание: | Процесс получения прогноза портфеля автострахования на основании анализа его структуры |
| Предусловия: | Пользователь включил компьютер |
| Постусловия: | Пользователь получает прогноз |
| Основной поток (нормальное течение): | 1. Пользователь запускает программу 2. Пользователь загружает исходные данные 3. Программа проверяет исходные данные 4. Точка расширения 5. Пользователь запускает один из алгоритмов анализа 6. Программа производит анализ данных 7. Пользователь получает результаты прогноза |
| Альтернативный поток (альтернативные течения): | 4a Программа проверяет корректность загруженных данных:  4a1a Программа обнаруживает, что пользователь загрузил неверные исходные данные- переход к пункту 2  4a1b Программа обнаруживает, что пользователь загрузил корректные данные - переходит к пункту 5 |

**5.5 ДИАГРАММА КОМПОНЕНТОВ**

Все рассмотренные ранее диаграммы относились к логическому уровню представления.

Особенность логического представления заключается в том, что оно оперирует понятиями, которые не имеют материального воплощения. Они лишь отражают наше понимание структуры программного обеспечения или аспекты ее поведения.

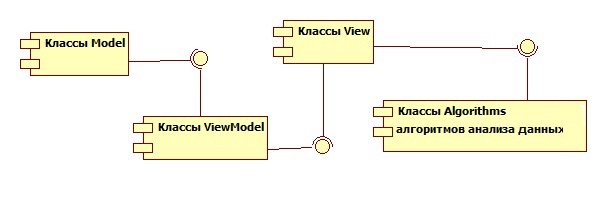
Однако для создания конкретной физической программы необходимо некоторым образом реализовать все элементы логического представления в конкретные материальные сущности. Для описания таких реальных сущностей предназначен другой аспект модельного представления, а именно физическое представление модели. Для физического представления моделей систем используются диаграммы реализации, которые включают в себя две отдельные диаграммы: диаграмму компонентов и диаграмму размещения.

Диаграмма компонентов отображает зависимости программных компонентов, которые представляются в виде исходных, откомпилированных и исполняемых программных кодов объектов. Один компонент, как правило, соответствует программному коду одного пакета классов объектов.

Компонент в своем составе имеет интерфейсный класс объектов, через который осуществляется доступ к остальным классам объектов компонента. На диаграмме интерфейс, предоставляющий данные, обозначается маленьким кружком, присоединенным к пиктограмме компонента. Интерфейс, получающий данные, обозначается маленькой вилочкой, присоединенной к пиктограмме компонента. С помощью интерфейса объекты других компонентов обращаются не к конкретным объектам рассматриваемого компонента, а к его интерфейсному объекту.

Таким образом, упрощается взаимодействие компонентов между собой, когда при доступе к компоненту из других компонентов не требуется знать внутреннюю структуру этого компонента. Компонент, к которому осуществляется обращение, может быть не объектно-ориентированным. Достаточно, чтобы у такого компонента был только один интерфейсный класс объектов, который транслирует запросы к компоненту в вызовы обычных процедур. У компонентов может быть несколько интерфейсов [7].

На рисунке 5.5 представлена диаграмма компонентов, на рисунке 5.6 – диаграмма размещения.

**** Рисунок 5.5 – Диаграмма компонентов

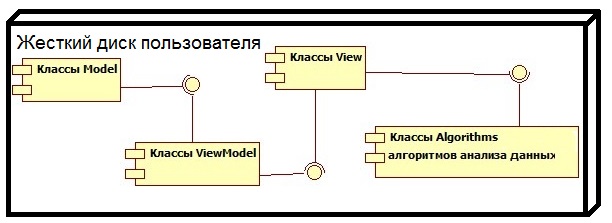


Рисунок 5.6 – Диаграмма размещения

1. **ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

**6.1 ОБЩИЙ АЛГОРИТМ РАБОТЫ ПРОГРАММЫ**

Обеспечивает ввод исходных данных, настройку алгоритмов анализа, выполнение анализа данных, вывод и сохранение полученных результатов (рисунок 6.1).

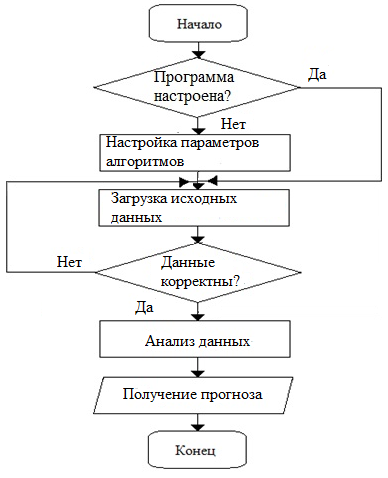
****

Рисунок 6.1 – Общий алгоритм работы программы

**6.2 СОСТАВ ПРОЕКТА**

Для разработки программной обеспечения была использована среда программирования MS Visual Studio 2013 Ultimate и язык программирования C#.

При реализации использовался принцип объектно-ориентированного программирования.

Проект содержит классы, окна, файлы ресурсов проекта. Листинг основных классов представлен в приложении.

Рассмотрим состав проекта созданного приложения подробнее.

*Основные файлы проекта:*

* App.xaml – определение стандартного приложения с помощью разметки языка XAML;
* App.xaml.cs – класс, содержащий связанный код для App.xaml;
* AboutWindow.cs – содержит описание внешнего вида окна «О программе»
* Command.cs – содержит описание класса Command, предназначенного для реализации ICommand интерфейса, предоставляющего доступ к методам классов ViewModel с использованием делегатов.
* SettingsModel.cs – Model-класс для окна настроек;
* SettingsViewModel.cs – ViewModel-класс для для окна настроек;
* SettingsWindow.xaml – содержит описание разметки окна настроек с помощья языка XAML;
* SettingsWindow.xaml.cs – класс, содержащий связанный код для SettingsWindow.xaml.
* MainWindow.xaml.cs – класс, содержащий связанный код для MainWindow.xaml.
* MainWindow.xaml.cs – класс, содержащий связанный код для MainWindow.xaml.
* CSVReader.cs – класс для считывания данных из файла.
* ConvertString.cs – класс для считывания данных из файла.
* CLOPE.cs – класс для алгоритма CLOPE.
* Kmeans.cs – класс для алгоритма k-средних.

Все вышеуказанные файлы находятся в папке InsPort\ InsPort.

*Системные файлы проекта:*

Системные файлы проекта генерирует среда разработки, они находятся в папке InstPort:

* InstPort.sln – файл решения;
* Settings.settings – файл настроек программы;
* файлы ресурсов Settings.Designer.cs и Resources.resx.

**6.3 СПЕЦИФИКАЦИЯ ВНУТРЕННИХ ФУНКЦИЙ**

Данный раздел содержит описание функционирования основных методов разработанного программного обеспечения. Текст данных методов приведен в приложении.

1. **Класс App.xaml.cs.**

В нем реализованы следующие методы:

* public App() – конструктор, предназначенный для инициализации программы.

1. **Класс AboutWindow.cs.**

В нем реализованы следующие методы:

* public About() – конструктор класса окна «О программе», предназначенный для инициализации компонентов;
* private void Cansel\_Click(object sender, RoutedEventArgs e) – метод, отвечающий за закрытие окна о программе.

1. **Класс Command.cs.**

В нем реализованы следующие методы:

* public Command(Action<object> action) – конструктор класса, предназначенный для инициализации;
* public bool CanExecute(object parameter) – метод, отвечающий за проверку возможности вызова делегата;
* public void Execute(object parameter) – метод, осуществляющий вызов необходимого делегата.

1. **Класс SettingsViewModel.cs .**

В нем реализованы следующие методы:

* public void Load() – метод, отвечающий за загрузку исходных данных из файла ресурсов.
* private void Update() – метод, отвечающий за обновление данных в файле ресурсов.

1. **Класс SettingsWindow.xaml.cs**

В нем реализованы следующие методы:

* public SettingsWindow() – конструктор окна настроек приложения.
* public void Save(object sender, RoutedEventArgs e) – метод, отвечающий за сохранение настроек.
* private void Cansel\_Click (object sender, RoutedEventArgs e) – метод, отвечающий за отмену изменения настроек и закрытие окна настроек.

**Класс MainWindow.xaml.cs**.

В нем реализованы следующие методы:

* public MainWindow() – конструктор главного окна приложения, предназначенный для инициализации компонентов.
* private void Guid\_Click(object sender, RoutedEventArgs e) – метод, отвечающий за запуск руководства пользователя;
* private void LoadData(object sender, RoutedEventArgs e) – метод, отвечающий за зугрузку исходных данных.
* private void AnalyseStart(object sender, RoutedEventArgs e) – метод, отвечающий за запуск процедуры анализа.
* private void Exit(object sender, RoutedEventArgs e) – метод, отвечающий за завершение работы приложения.

**6.4 СПЕЦИФИКАЦИЯ ДАННЫХ**

Входными данными для программного обеспечения является файл в формате .csv с информацией о заключенных договорах страховой компании, со следующими данными по столбцам, отраженными в таблице 6.1.

Таблица 6.1 – Входные данные

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Х1 | Х2 | Х3 | Х4 | Х5 | Х6 | Х7 | Х8 | Х9 |

где:

№– номер по списку;

X1 – стоимость автомобиля;

X2 – средний убыток по объекту;

X3 – срок эксплуатации;

X4 – износ;

X5 – мощность;

X6 – масса автомобиля;

X7 – возраст водителя;

X8 – стаж водителя;

X9 – риск ДТП;

Информация вводиться вручную, путем загрузки файла исходных данных, при помощи управляющих компонентов главной формы приложения.

Выходными данными по результатам работы программы является полученный прогноз, который, на выбор пользователя, может быть сохранен с помощью табличного процессора Excel или распечатан на принтере.

1. **ПРОГРАММНАЯ ДОКУМЕНТАЦИЯ**
   1. **НАЗНАЧЕНИЕ ПРОГРАММЫ**

Программа предназначена для анализа структуры портфеля автострахования физических лиц.

Исходными данными является упорядоченная информация о заключенных договорах страховой компании.

Разработанное программное обеспечение осуществляет анализ исходных данных и прогноз показателей портфеля.

При этом программа реализует табличное представление результатов в виде списка прогнозных значений тарифной политики, предоставляет возможность выгрузки полученных данных в Excel или вывод их на печать.

Разработанная программа может применяться страховыми компаниями и другими профессиональными участниками рынка автострахования, для автоматизации анализа большого количества информации и удобного представления результатов с прогнозом показателей, что в свою очередь упрощает принятие решений о выборе той или иной тарифной политики.

* 1. **Требования к составу и параметрам технических средств**

Система предназначена для работы на персональном компьютере (ПК). ПК должен удовлетворять следующим требованиям:

* на ПК должна быть установлена операционная система Windows ХР/7/8 и платформа Microsoft .NET Framework 3.5;
* ПК должен удовлетворять минимальным системным требованиям установленной операционной системы (различаются в зависимости от версии Windows);
* минимальное допустимое разрешение монитора 1024х768, видеокарта и монитор должны поддерживать этот режим работы;
* манипулятор типа «мышь»;
* клавиатура.
  1. **Руководство пользователя**

Запуск программы производится либо при загрузке файла IntPort.exe из папки IntPort, либо двойным щелчком на ярлыке программы на рабочем столе пользователя.

Для создания на рабочем столе ярлыка программы необходимо выделить файл IntPort.exe правым щелчком мыши и выбрать пункт Отправить/Рабочий стол (создать ярлык).

После этого на рабочем столе будет создан ярлык программы IntPort.

После запуска приложения на экране появляется главное окно программы и откроется вкладка «Приветствие» (рисунок 7.1):

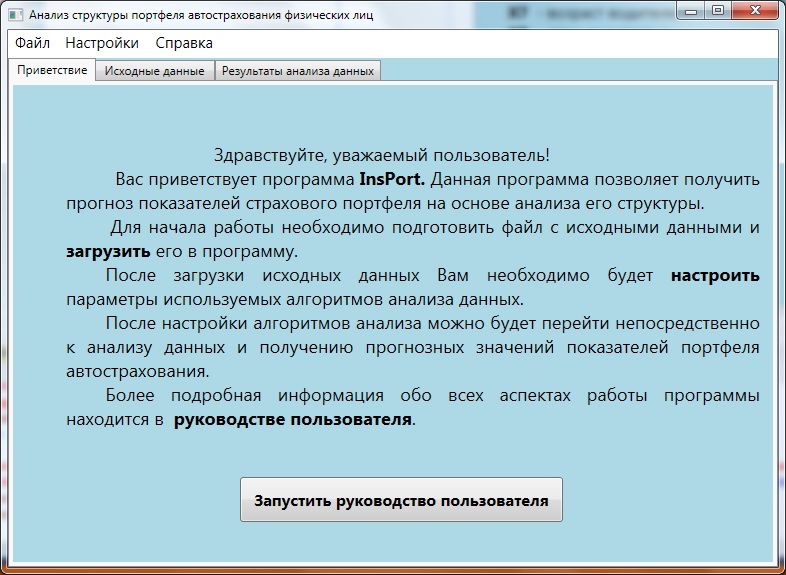


Рисунок 7.1 – Главное окно программы

В верхней части окна представлено главное меню программы (рисунок 7.2):

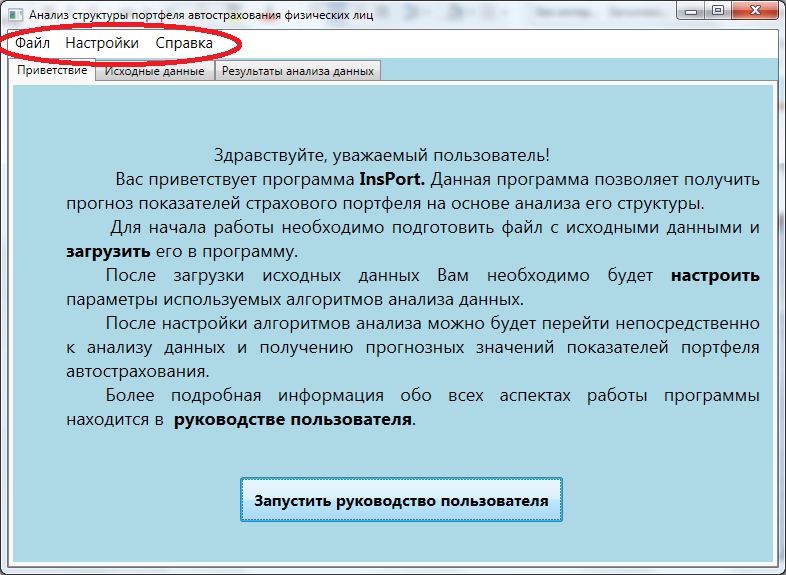


Рисунок 7.2 – Главное меню программы

Для открытия окна настроек необходимо выбрать пункт меню *Настройки ->Настройки алгоритмов* (рисунок 7.3):

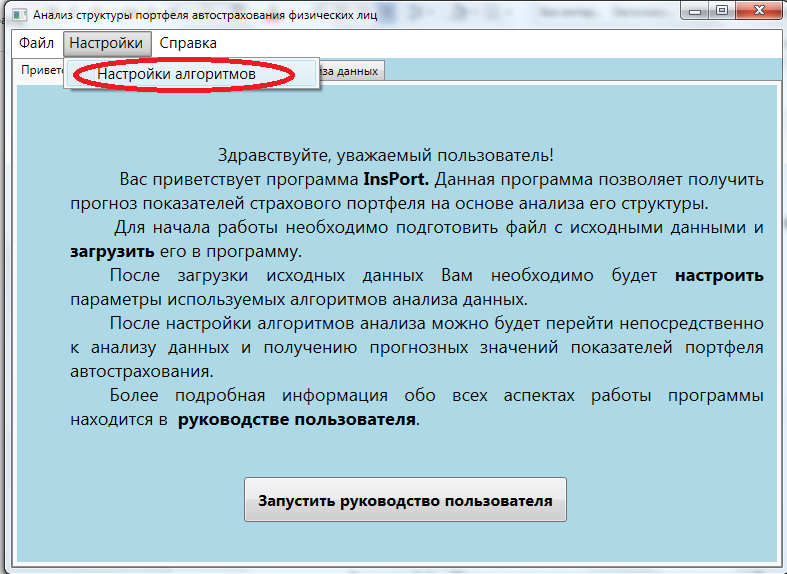


Рисунок 7.3 – Пункт меню для перехода к окну настроек

Откроется окно настроек (рисунок 7.4):

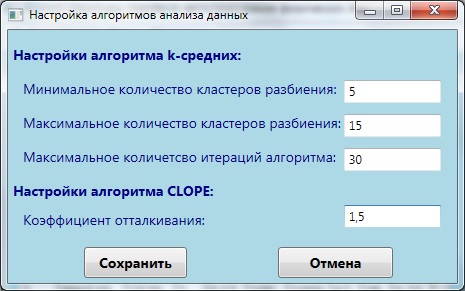


Рисунок 7.4 – Окно настроек программы

В настройках можно поменять основные параметры используемых алгоритмов. Для сохранения настроек необходимо нажать кнопку «Применить» (рисунок 7.5):

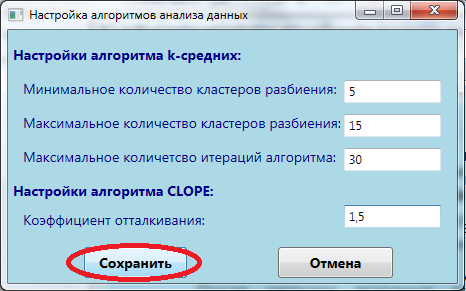


Рисунок 7.5 – Применение настроек

Для вызова руководства пользователя необходимо выбрать пункт главного меню *Справка->Руководство пользователя* (рисунок 7.6):

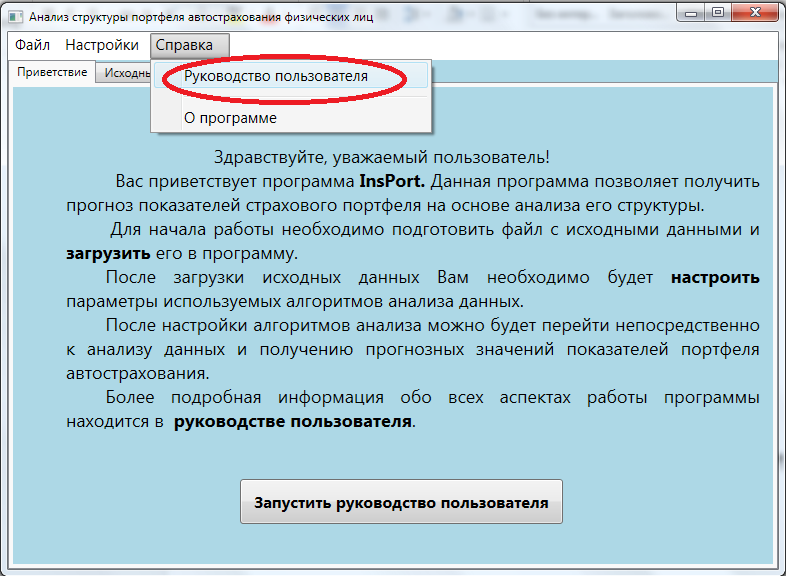


Рисунок 7.6 – Выбор пункта меню «Руководство пользователя»

Откроется руководство пользователя (рисунок 7.7):

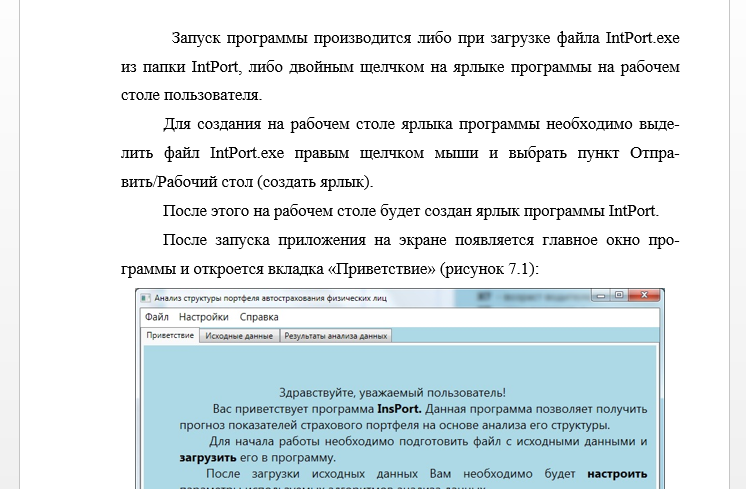


Рисунок 7.7 – Запущенное руководство пользователя в программе для просмотра .pdf файлов

Далее следует загрузить исходные данные, для этого необходимо выбрать пункт главного меню *Файл->Загрузка исходных данных* и выбрать подготовленный файл (рисунок 7.8):

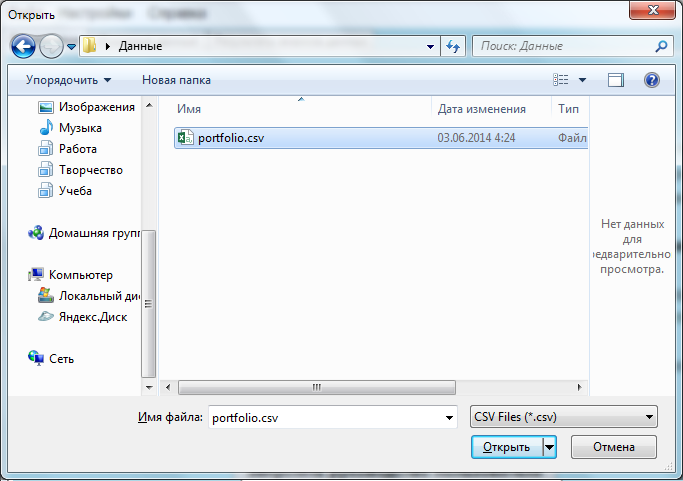


Рисунок 7.8 – Выбор файла с исходными данными

Файл успешно загрузится (рисунок 7.9):

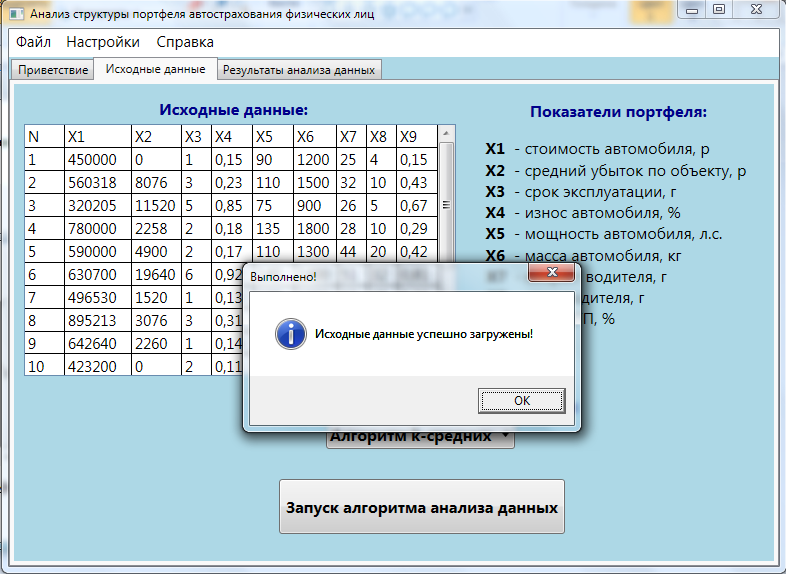


Рисунок 7.9 – Успешная загрузка файла с данными

Далее следует перейти непосредственно к анализа данных. Для этого необходимо выбрать в списке один из алгоритмов анализа и нажать на кнопку «Запуск алгоритма анализа данных» на вкладке «Исходные данные» (рисунок 7.10)

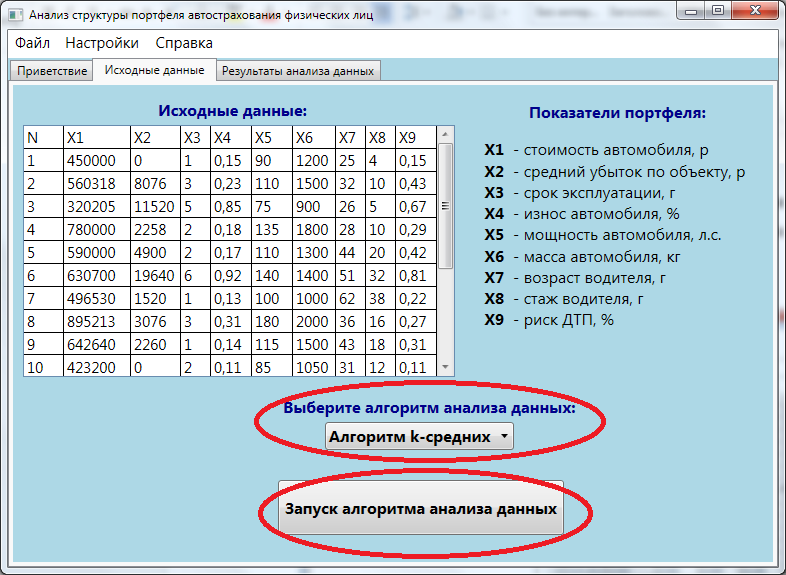


Рисунок 7.10 – Выбор алгоритма и нажатие на кнопку «Запуск алгоритма анализа данных» для запуска процедуры анализа

По окончанию процедуры анализа, пользователю в наглядном виде будут представлены результаты (рисунок 7.11):

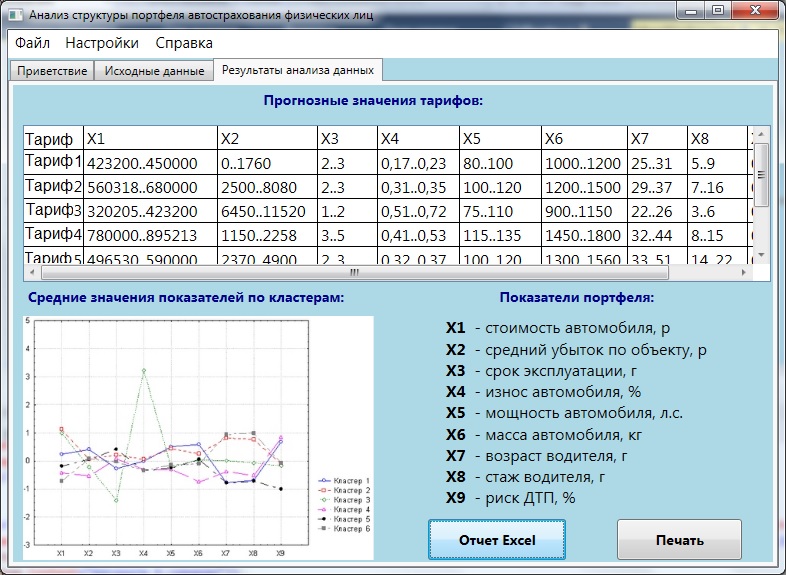


Рисунок 7.11 – Результаты анализа данных

Полученные данные пользователь может сохранить одним из двух способов: при нажатии на кнопку «Отчет Excel» результаты прогноза автоматически сохранятся в табличном редакторе, а при нажатии на кнопку «Печать» - автоматически выведутся на принтер (рисунок 7.12):

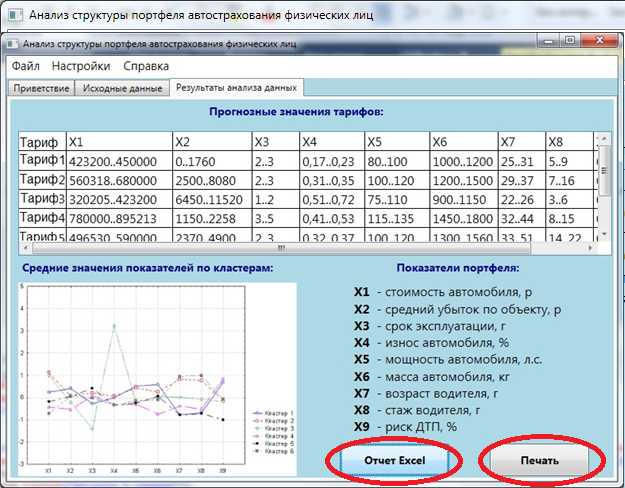


Рисунок 7.12 – Выбор способа сохранения полученных результатов

Для выхода из программы необходимо выбрать пункт главного меню *Файл->Выход* (рисунок 7.13):

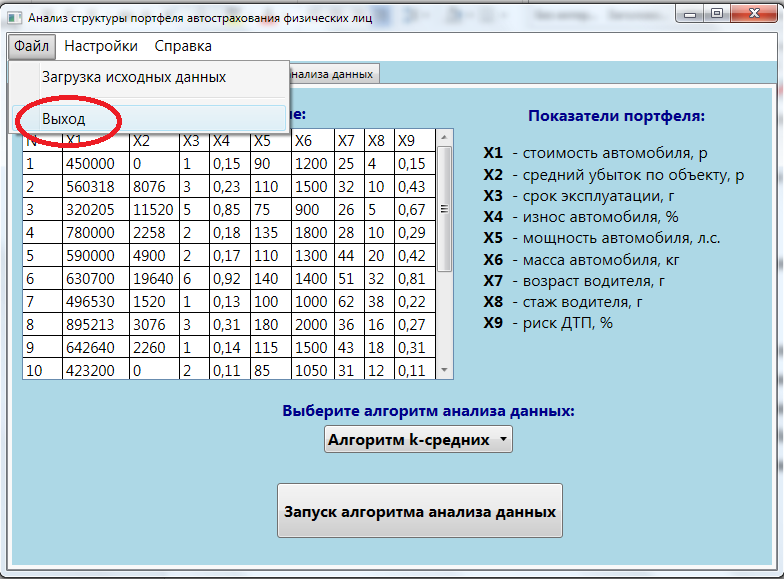


Рисунок 7.13 – Выбор пункта главного меню «Выход» для завершения работы с программой

* 1. **Руководство программиста**

*Характеристика программы.*

Данная программа анализа структуры портфеля автострахования написана на языке программирования C#. В качестве среды разработки была использована среда программирования MS Visual Studio 2013 Ultimate.

При проектировании были использованы основные подходы объектно-ориентированного программирования, поэтому состав программного проекта входят отдельные классы (см. п. 6.2).

*Обращение к программе.*

Для корректной инсталляции разработанного приложения необходимо скопировать на жесткий диск папку «InsPort», содержащую исполняемый файл программы InsPort.

Для создания на рабочем столе ярлыка программы необходимо выделить файл InsPort.exe правым щелчком мыши и выбрать пункт Отправить/Рабочий стол (создать ярлык). После этого на рабочем столе будет создан ярлык программы InsPort.

Для старта программы необходимо запустить файл InsPort.exe.

Дальнейшее обращение к программе сводится к работе с ней как со стандартной программой, написанной для операционных систем семейства Windows.

1. **ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**
   1. **ОБЪЕКТ ИСПЫТАНИЙ**

Объектом испытаний является программное обеспечение для прогнозирования структуры портфеля автострахования. Оно предназначено для автоматизации анализа структуры портфеля и получения прогноза показателей.

* 1. **ЦЕЛЬ ИСПЫТАНИЙ**

Целью испытаний является выявление различного рода ошибок и сбоев в работе программного обеспечения, чтобы обеспечить правильное и более надежное функционирование.

* 1. **ТРЕБОВАНИЯ К ПРОГРАММЕ**

При тестировании программного обеспечения необходимо проверить: правильность функционирования; адекватность действий всех управляемых элементов интерфейса. Программное обеспечение должно функционировать: при отсутствии или неверном формате входных данных; при заведомо неправильных действиях пользователя.

* 1. **СРЕДСТВА ИСПЫТАНИЙ**

При проведении испытаний программы использовалась следующая конфигурация аппаратных и программных средств.

Компьютер:

* Intel (R) Core(TM)2 QUAD CPU 2.4 ГГц;
* 3072 Мб ОЗУ.

Программные средства:

* ОС Windows 7 Ultimate SP1;
  1. **ПОРЯДОК ИСПЫТАНИЙ**

При тестировании программного обеспечения требуется несколько различных типов тестирования, таких как функциональное тестирование, тестирование интерфейса пользователя, тестирование производительности и конфигурационное тестирование. Каждый из этих типов тестов фокусируется на одной или нескольких целях.

* 1. **ИСПЫТАНИЯ ПРОГРАММЫ**

В процессе разработки программного обеспечения было использовано регрессионное нисходящее тестирование.

Сущность регрессионного тестирования заключается в том, что разработанные тесты многократно и полностью используются по ходу проекта с их обновлением и накоплением. Главная задача регрессионного тестирования заключается в отслеживании того, чтобы однажды выявленная и исправленная ошибка не повторилась в дальнейшем.

Сущность нисходящего тестирования заключается в том, что сначала тестируется программа в целом, а затем проверяются отдельные модули.

Так же проводились тесты по выявлению функциональных ошибок, ошибок пользовательского интерфейса, ошибок производительности и др. В результате в системе были устранены все функциональные ошибки, интерфейс пользователя приведен к максимально удобному и понятному для пользователя виду, практически полностью исключены ситуации некорректно обрабатываемых ошибок, полученных из-за неправильной работы с системой. Разработано руководство пользователя, содержащее подробные рекомендации, замечания и советы по наиболее оптимальному использованию программного обеспечения.

В рамках данной пояснительной записки, далее будет описано проведение стохастического тестирования основной функциональности программного обеспечения.

Сущность стохастического тестирования заключается в том, что проводится выборочная проверка программы – по случайной представительной выборке тестов. Оно проводится в том случае, когда полное регрессионного тестирование проводить нецелесообразно (по соображениям экономии времени; при тиражировании программы – каждый экземпляр полностью проверять слишком трудоемко и т.п.).

Проведем тестовый запуск программы:

* 1. Запустим программу (рисунок 8.1):

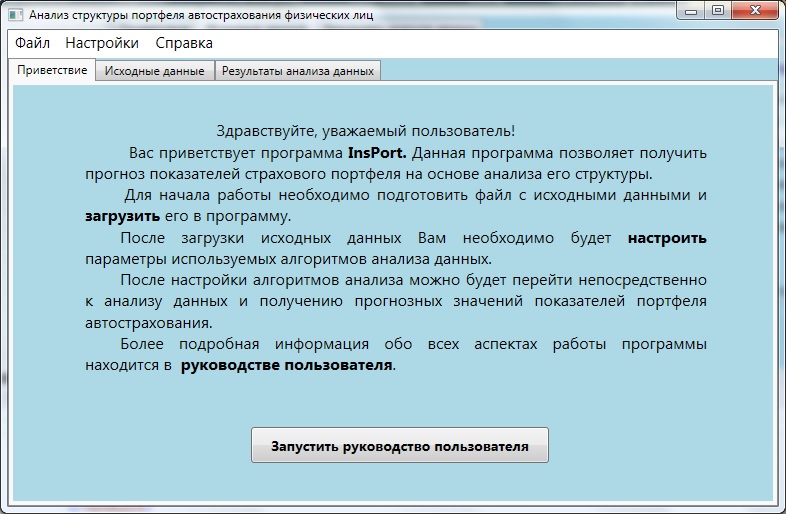


Рисунок 8.1 – Главное окно программы

1. Загрузим исходные данные (рисунок 8.2):

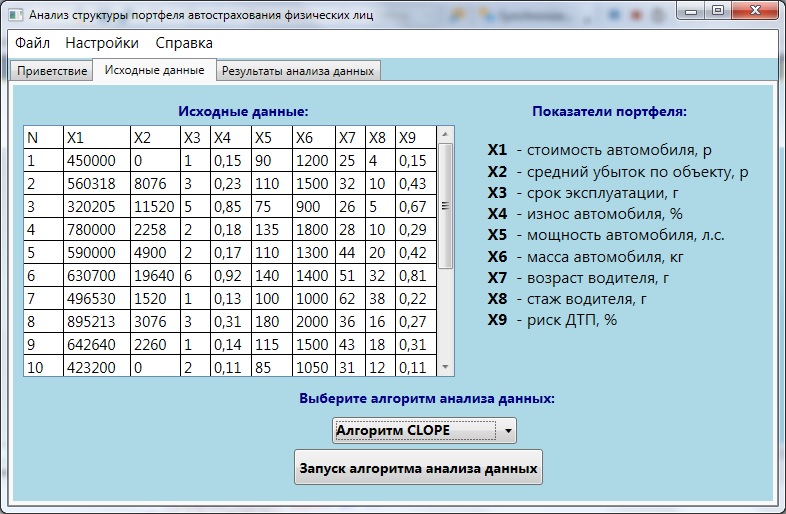


Рисунок 8.2 – Главное окно программы с загруженными данными

1. Зададим настройки алгоритмов анализа данных (рисунок 8.3):

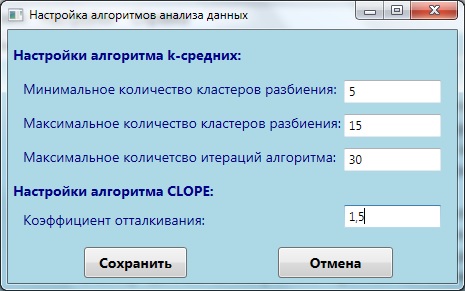


Рисунок 8.3 – Настройка алгоритмов анализа данных

1. Запустим один из алгоритмов и выведем график и результаты анализа (рисунок 8.4):

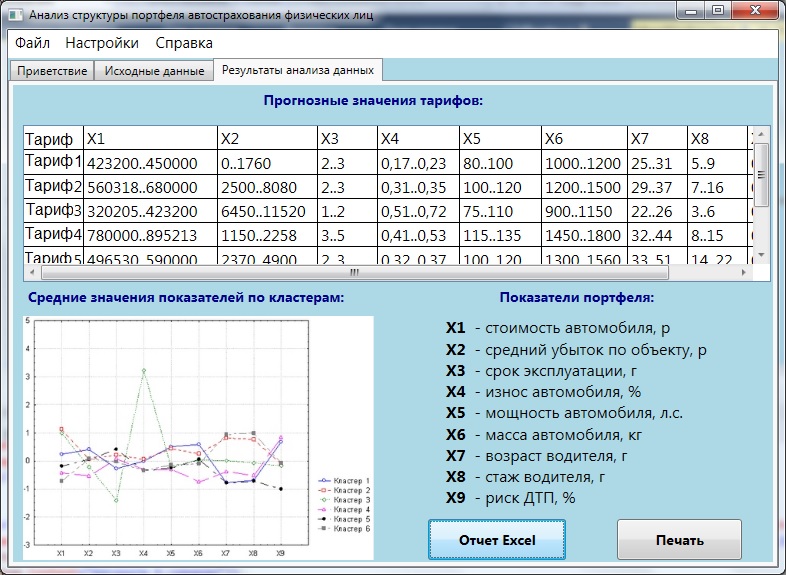


Рисунок 8.4 – Результаты анализа данных

Стоит отметить, что на всех этапах работы производится проверка ввода данных, и, в случае некорректного ввода, выдается сообщение об ошибке (рисунок 8.5):

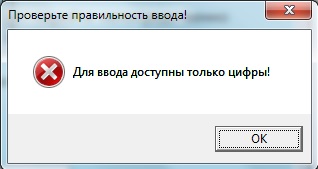


Рисунок 8.5 – Сообщение о некорректном вводе данных

По результатам тестов можно сделать вывод о работоспособности всех функций программы.

1. **ЭКОНОМИЧЕСКАЯ ЧАСТЬ**

Оценка экономической целесообразности проекта является неотъемлемой частью процесса разработки программного обеспечения.

Проанализируем разрабатываемую в дипломном проекте модель с экономической точки зрения. Результат анализа должен представлять собой доказательство целесообразности и экономической эффективности проводимого в рамках дипломного проекта исследования и подтверждать практическую применимость готового программного продукта.

9.1 ТЕХНИКО-ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ

Разработка программного обеспечения для ЭВМ требует в настоящее время больших трудовых и финансовых затрат, тем более, что сроки разработки программного продукта должны быть достаточно короткими для того, чтобы разработанное программное средство не успело «устареть». И эти затраты, естественно, тем больше, чем меньше срок разработки. Следовательно, разрабатывая программное обеспечение, необходимо уметь анализировать, оценивать и планировать трудовые и финансовые затраты для более эффективной разработки.

Программное обеспечение, автоматизирующее анализ структуры портфеля автотранспортного страхования физических лиц является актуальной для рынка автострахования, так как оно дает возможность прогнозировать показатели страхового портфеля в случае изменения его структуры. Что ведет к своевременному выявлению неэффективности портфеля.

Это в свою очередь позволяет минимизировать риски, поскольку при отсутствии своевременных сведений возможно принятие убыточного решения тарифного управления.

Главное внимание уделяется рассмотрению программного обеспечения с позиции использования ее возможностей для повышения эффективности труда специалистов страхования и поддержки принятия решений об управлении тарифами.

Программное обеспечение позволяет анализировать информацию о факторах, отражающих состояние и условия эксплуатации застрахованных объектов, и составлять прогноз их изменения, который помогает:

* повысить эффективность принятия решений тарифной политики
* экономить время
* снизить возможность возникновения ошибок, способных привести к убыткам.

Понятное и удобное приложение обеспечивает легкую работу с информацией. Система проста в освоении и не требует от пользователей специальных знаний.

На данный момент на рынке существуют лишь некоторые методики решения задачи анализа страхового портфеля, реализуемые лишь в специальных статистических программах (например, таких как Statistica), но комплексное специализируемое программное обеспечение отсутствует, что является основанием для разработки данного программного продукта.

Можно сделать вывод об экономической обоснованности разрабатываемого программного обеспечения и его востребованности на рынке.

## 9.2 ПЛАНИРОВАНИЕ ПОЭТАПНОЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Важным этапом разработки проекта является составление плана выполнения работ, определение их ресурсоемкости, стоимости и проведение оценки экономической эффективности его применения. Успешное решение сложного комплекса задач при разработке программного обеспечения возможно лишь при увязке всех работ по времени, исполнителям и ресурсам.

Для того чтобы наиболее полно составить план подготовки выполнения проекта, необходимо этапы максимально детализировать, выбрать такое направление для воздействия на ход создания этой системы, чтобы весь комплекс работ был выполнен в кратчайшие сроки и с минимальными затратами.

Процесс разработки данного дипломного проекта можно разделить на ряд этапов, каждый из которых состоит из определенного перечня работ. Одной из основных задач является определение продолжительности каждой работы для последующего определения общей продолжительности всего проекта. Это можно обеспечить на основе составления ленточного графика. Ленточный график отражает длительность и последовательность выполняемых работ. Для составления ленточного графика учитываются все виды работ, проделанные в процессе дипломного проектирования.

Разработка программного продукта была проведена в 2014 г. в период 01.03 – 15.04, что составляет 46 календарных дня. На основании полученных результатов можно спланировать длительность этапов разработки в соответствии с типовыми этапами.

В данном дипломном проекте были проведены следующие работы:

* разработка технического задания;
* сбор и изучение исходных данных;
* разработка ПО;
* тестирование и отладка ПО;
* разработка программной документации;
* оформление пояснительной записки;
* оформление графического материала.

Составим таблицу, содержащую план проектирования программно-информационной системы по стадиям разработки. Трудоемкость выполнения всей проектной разработки определяется по сумме трудоемкости этапов и видов работ. Экспертным путем была оценена трудоемкость *Тi* отдельных этапов работы в человеко-днях. Данная продолжительность показывает сколько дней потребуется одному человеку для выполнения определенного этапа работ [3].

Длительность выполнения каждого этапа в днях (Tn) определяем как отношение трудоемкости этапа к количеству исполнителей:

 (9.1)

где *Ti* - трудоемкость работ, человеко-дни;

*ni* - численность исполнителей, человек.

Распределение исполнителей по этапам работ, продолжительность работ и результаты расчета продолжительности выполнения проекта приведены в таблице 9.1.

Таблица 9.1.

План проектирования программной продукции по стадиям разработки

| **№**  **этапа** | **Стадии разработки проекта** | **Исполнители** | ***Тi*, *чел/***  ***дни*** | **Кол-во**  **человек, *n*** | **Продолжительность, *Tn=Ti/n, дни*** |
| --- | --- | --- | --- | --- | --- |
| 1 | Разработка технического задания | Руководитель, студент | 4 | 2 | 2 |
| 2 | Сбор и изучение исходных данных | Руководитель, студент | 14 | 2 | 2 |
| 3 | Разработка программы | Студент | 26 | 1 | 26 |
| 4 | Тестирование и отладка приложения | Студент | 4 | 1 | 4 |
| 5 | Разработка программной документации | Студент | 6 | 1 | 6 |
| 6 | Оформление пояснительной записки | Руководитель, студент | 18 | 2 | 4 |
| 7 | Оформление графического материала | Студент | 2 | 1 | 2 |
|  | Итого |  | 54 |  | 46 |

Ленточный график на рис. 9.1 построен на основе перечня выполняемых работ, приведенных в таблице 9.1. График отражает работы, которые являются наиболее существенными в отношении данного дипломного проекта.

По вертикальной оси расположены номера этапов, по горизонтальной оси – сроки начала и завершения этапов работ.

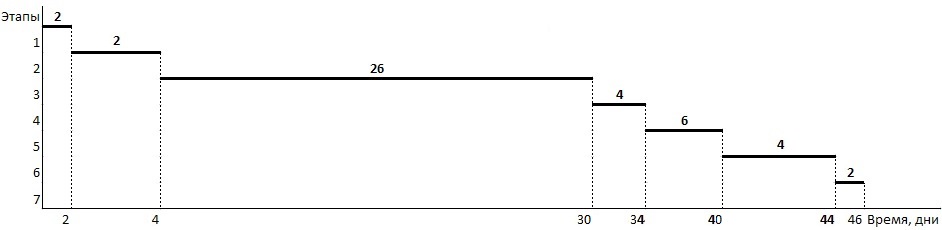


Рисунок 9.1 - Ленточный график

Выполненные расчеты и построенный ленточный график показывают, что суммарная продолжительность проекта по созданию программного продукта составит 46 день, следовательно, проект укладывается в отведенные временные сроки [3].

## 9.3 СОСТАВЛЕНИЕ СМЕТЫ ЗАТРАТ НА РАЗРАБОТКУ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Затраты на разработку ПО представляет собой стоимостную оценку материалов, энергии, основных фондов, трудовых ресурсов, а также других затрат на их производство и реализацию, используемых в процессе создания.

Затраты, образующие себестоимость продукции группируются в соответствии с их экономическим содержанием по следующим элементам:

* материальные затраты;
* затраты на оплату труда;
* амортизация основных фондов;
* прочие затраты.

**9.3.1 МАТЕРИАЛЬНЫЕ ЗАТРАТЫ**

Специальное оборудование и ПО для разработки проекта не приобреталось, поэтому в материальные затраты включаем только расходы на электроэнергию:

,

где – потребляемая мощность оборудования, кВт/ч; – стоимость 1 кВт/ч, руб.; – время использования оборудования при проведении работ, ч.

Для выполнения работы использовался персональный компьютер потребляемой мощностью 360 Вт и принтер потребляемой мощностью 350 Вт. Время работы ПЭВМ составляет 46 дней по 8 часов в день, а принтера – 4 часа. Стоимость 1 кВт для РГРТУ на 2014 г. – 3,1 руб./кВт.

Исходя из приведенных данных, рассчитаем :



Следовательно, получаем, что материальные затраты составляют 415,05 рублей.

**9.3.2 ЗАТРАТЫ НА ОПЛАТУ ТРУДА**

Затраты на оплату труда начисляются исходя из ставок разработчика и руководителя проекта, а также времени затрачиваемого на выполнение работы.

Затраты на оплату труда включают:

* выплаты заработной платы за фактически выполненную работу, исчисленные исходя из сдельных расценок, тарифных ставок, должностных окладов;
* выплаты стимулирующего характера, связанные с режимом работы, условиями труда;
* оплата очередных и дополнительных отпусков, льготных часов подростков;
* другие виды доплат, предусмотренные законами РФ и включаемые в фонд оплаты труда.

Оклад руководителя от кафедры составляет 10000 рублей. Окладом разработчика (студента-дипломника) является его стипендия и составляет 0 рублей, так как студент обучается на платной основе.

Таким образом, исходя из затрат времени на разработку (руководитель – 16 дней, разработчик – 46 дней), заработная плата равна:

.

.

Итого общий фонд заработной платы составляет:



**9.3.3 АМОРТИЗАЦИОННЫЕ ОТЧИСЛЕНИЯ**

Под амортизацией основных средств и фондов предприятия понимается периодическое уменьшение и списание их первоначальной стоимости, вплоть до окончательного списания с баланса предприятия. Это необходимо по причине естественного старения и износа основных фондов организаций и предприятий. Кроме того, на каждый агрегат или технику, даже мелкую офисную, предусмотрен срок эксплуатации. На его основе, в бухгалтерском учете предприятия закладывается срок использования данной единицы, и рассчитывается амортизация основных фондов.

Согласно 258 статье НК РФ амортизируемым имуществом признается имущество со сроком полезного использования более 12 месяцев и первоначальной стоимостью более 40 000 рублей. В дипломном проекте объектом основных фондов будет являться персональный компьютер.

Расчет амортизационных отчислений производится по следующей формуле:

,

где  – балансовая стоимость оборудования;  – время использования оборудования при проведении работ;  – норма амортизации;  – годовой эффективный фонд времени работы оборудования, для односменной работы он составляет .

,

где  – срок службы оборудования, лет.

Согласно классификации основных средств, включаемых в амортизационные группы, компьютер входит во вторую амортизационную группу, исходя из этого, срок его полезного использования составляет 2-3 года, для проведения дальнейших расчетов установим срок полезного использования равный 3 года. Тогда норма амортизации составит:



Время работы на ПЭВМ составляет 46 дней. Амортизационные отчисления для компьютера стоимостью в 40000 рублей составят:



Общие прямые затраты составят следующую сумму:



**9.3.4 ПРОЧИЕ РАСХОДЫ**

Для того, чтобы определить величину прочих расходов, необходимо найти сумму величины страховых взносов и остальных прочих расходов.

В 2014 году страховые взносы берутся в размере 30,2 % от величины фонда оплаты труда. Рассчитаем величину страховых взносов:



Величина остальных прочих расходов берется от суммы прямых общих затрат в установленном размере (20%) и может быть рассчитана по формуле:



Подставив свои значения, получим:



Прочие расходы составят:



Общие затраты на разработку составят:



Представим полученные данные по статьям сметы затрат в таблице 8.2.

Таблица 8.2 – Смета расходов

|  |  |  |
| --- | --- | --- |
| **Наименование калькуляционных статей расходов** | **Сумма, руб.** | **Удельный вес, %** |
| Материальные затраты, | 415,03 | 2,84 |
| Затраты на заработную плату, | 7272,72 | 49,72 |
| Амортизация оборудования, | 2371,86 | 16,21 |
| Прочие расходы, | 4568,23 | 31,23 |
| **Общие затраты,** | **14627,84** | **100** |

Смета затрат на разработку проекта составила 14627,84 рублей.

**9.3.5 РАСЧЕТ ЦЕНЫ НИР**

Устанавливая цену на программный продукт, нужно исходить из необходимости компенсации затрат на его производство, уплаты государству налогов и получение прибыли для дальнейшего развития.

Состав расчётной цены на научно-исследовательскую разработку определяется следующей формулой:

,

где  – затраты на разработку;  – прибыль от реализации.

Определим расчётную цену на продукцию при предполагаемом размере прибыли в 10%:



.

.

## 9.4 ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ

Оценить экономическую эффективность применения разработанного программного обеспечения в числовом выражении достаточно трудно. Это связанно с тем, что ее эффективность будет складываться из множества составляющих, причем, некоторые из них практически не поддаются числовой оценке.

Система в первую очередь ориентирована на улучшение качества работы специалистов автострахования и оценить прямой экономический эффект потребителя системы от ее внедрения представляется затруднительным.

Для производителя системы основным показателем эффективности проекта будет являться прибыль, полученная в ходе реализации копий программы. Реальная прибыль от реализации программного продукта рассчитывается по формуле:



где Цр – цена свободная рыночная; Сз – сумма затрат на разработку; N – число проданных копий; Стир – затраты на тиражирование.

Затраты на тиражирование включают:

стоимость CD-диска – 13 руб, без НДС – 10,66 руб.

стоимость копирования – 20 % от стоимости диска без НДС – 2,13 руб.

Общие затраты на тиражирование: Стир= 10,66 + 2,13 = 12,79 руб.

С учетом всех особенностей данная программа может хорошо продаваться по рыночной цене приблизительно в 3-10 тыс. рублей.

Установим свободную рыночную цену одной копии программы в 5 000 руб. и проведем расчет зависимости реальной прибыли от числа проданных копий. Данные расчета приведены в таблице 8.3.

Таблица 8.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Количество, копии | 1 | 2 | 3 | 4 | 5 |
| Прибыль, руб. | -10403,3 | -6178,84 | -1954,35 | 2270,153 | 6494,651 |

Диаграмма зависимости реальной прибыли от количества проданных копий представлена на рисунке 9.2:

Рисунок 9.2 – Зависимость реальной прибыли от количества проданных копий

Расчеты показывают, что реализация не менее 4 копий программы принесет реальную положительную прибыль.

Экономия для пользователя от проекта заключается в получении экономии времени, за счет отсутствия необходимости оценивать показатели страхового портфеля (порядка 10 показателей по почти 1800 договоров страхования, что составит около 20000 вычислительных операций, результат которых необходимо каждый сопоставить друг с другом), так как данная автоматизация является главной причиной разработки данного программного продукта.

Необходимо отметить, что проект является клиентоориентированной разработкой, так как он легок в освоении, в нем простой и удобный интерфейс, что важно для комфортного ежедневного использования.

По итогам экономической части можно сделать вывод: разработка программного обеспечения экономически обоснована и целесообразна. Основное назначение программного продукта заключается в автоматизации и упрощении работы специалистов рынка автострахования, с его помощью они смогут оперативно получить прогноз показателей страхового портфеля в случае изменения его структуры для принятия оптимальной стратегии управления тарифной политикой.

Разрабатываемое программное обеспечение отличает приемлемая цена, дружественный интерфейс, соответствие требованиям потенциальных потребителей, отсутствие необходимости изучать сложные статистические пакеты и навыки работы с ними, так как данный продукт реализует все необходимые возможности.

Таким образом, создание и внедрение программного обеспечения является экономически выгодным проектом, как для разработчика, так и для ее потенциального потребителя.

1. **БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ ПРОЕКТА**

Темой дипломного проекта является разработка программного обеспечения для рабочего места оператора ПЭВМ. Работа с компьютером характеризуется значительным умственным напряжением и нервно-эмоциональной нагрузкой операторов, высокой напряженностью зрительной работы и достаточно большой нагрузкой на мышцы рук при работе с клавиатурой ПЭВМ.

Проанализируем рабочее место инженера на наличие вредных и опасных факторов, влияющих на его работу.

## 10.1 ОПИСАНИЕ РАБОЧЕГО ПОМЕЩЕНИЯ, РАБОЧЕГО МЕСТА, ВЫПОЛНЯЕМЫХ ОПЕРАЦИЙ

Рассмотрим воздействие опасных и вредных факторов в помещении, где находится рабочее место пользователя ПЭВМ.

Естественное и искусственное освещение соответствует требованиям действующей нормативной документации. Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно ориентированы на север и северо-восток. Оконные проемы оборудованы регулируемыми устройствами, а именно жалюзи.

План помещения, в котором проводилось дипломное проектирование представлен на рисунке 10.1, где 1 – оконный проем,   
2 – дверной проем.

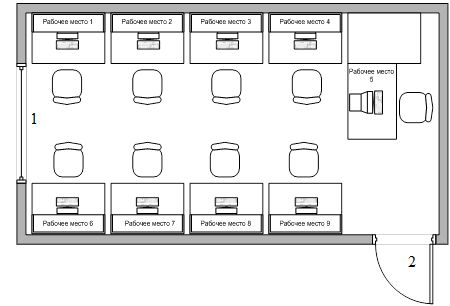


Рисунок 10.1 – План помещения

Размеры помещения – 5,6x8,3 м. Площадь помещения 46,48 м2. Высота – 3,5 м. Таким образом, на каждое рабочее место приходится 4,5 м2, что соответствует требованиям СанПиН 2.2.2.542-96. Ширина оконного проема 1,5 м. Размеры столов - 1\*0,8 м. Кроме того, учтено требование о размещении рабочих мест таким образом, чтобы естественное освещение падало на мониторы сбоку и слева.

На рабочем месте присутствует следующее оборудование:

* монитор-(ЖК);
* клавиатура;
* манипулятор типа «мышь»;
* лазерный принтер.

В рассматриваемом помещении поддерживаются следующие микроклиматические условия:

* средняя температура воздуха – 21С (в холодное время года – 19 С, в теплое – 23С).
* относительная влажность – около 40% – 70%, в зависимости от погоды на улице.
* движения воздуха от 0.1 м/с при безветренной погоде, до 0,3 м/с при ветре (основная причина по которой увеличивается скорость ветра в помещении – это отсутствие герметичности оконной коробки).

В помещении отопление центральное водяное, четыре лампы дневного света, кондиционер. Из горючих материалов присутствует бумага, картон, дерево, обмотка проводов. Оборудование питается от сети переменного тока напряжением 380/220В, с частотой 50 Гц и глухозаземленной нейтралью.

Общие требования к организации рабочих мест пользователей ПЭВМ определяются в соответствии с СанПиН 2.2.2/2.4.1340-03.

Экран видеомонитора должен находиться от глаз пользователя на расстоянии 600 - 700 мм, но не ближе 500 мм с учетом размеров алфавитно-цифровых знаков и символов.

Конструкция рабочего стола должна обеспечивать оптимальное размещение на рабочей поверхности используемого оборудования с учетом его количества и конструктивных особенностей, характера выполняемой работы. При этом допускается использование рабочих столов различных конструкций, отвечающих современным требованиям эргономики.

Оптимальный план организации рабочего места в соответствии с требованиями представлен на Рисунке 10.2:

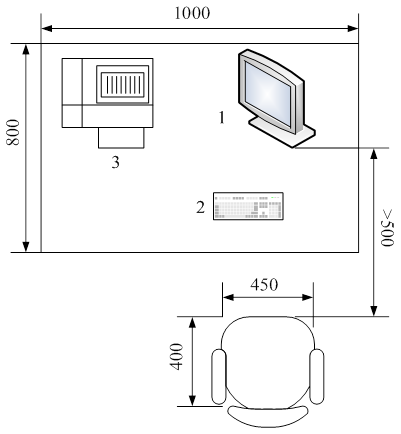


Рисунок 10.2: – Организация рабочего места.

На Рисунке 10.2 цифрами обозначены:

1 – монитор (ЖК)

2 – клавиатура

3 – принтер

Системный блок компьютера в связи с редким его использованием и значительными габаритами рекомендуется устанавливать в специальную нишу внутри стола. Также на столе остается достаточно места для работы с документами. Поверхность рабочего стола должна иметь коэффициент отражения 0,5 – 0,7.

Конструкция рабочего стула (кресла) должна обеспечивать поддержание рациональной рабочей позы при работе на ПЭВМ позволять изменять позу с целью снижения статического напряжения мышц шейно-плечевой области и спины для предупреждения развития утомления. Тип рабочего стула (кресла) следует выбирать с учетом роста пользователя, характера и продолжительности работы с ПЭВМ.

Рабочий стул (кресло) должен быт подъемно-поворотным, регулируемым по высоте и углам наклона сиденья и спинки, а также I расстоянию спинки от переднего края сиденья, при этом регулировка каждого параметра должна быть независимой, легко осуществляемой и иметь надежную фиксацию.

Высота рабочей поверхности стола для взрослых пользователей должна регулироваться в пределах 680 - 800 мм; при отсутствии такой возможности высота рабочей поверхности стола должна составлять 725 мм. Модульными размерами рабочей поверхности стола для ПЭВМ, на основании которых должны рассчитываться конструктивные размеры, следует считать: ширину 800, 1000, 1200 и 1400 мм, глубину 800 и 1000 мм при нерегулируемой его высоте, равной 725 мм. Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной не менее 500 мм, глубиной на уровне колен - не менее 450 мм и на уровне вытянутых ног - не менее 650 мм.

Конструкция рабочего стула должна обеспечивать: ширину и глубину поверхности сиденья не менее 400 мм; поверхность сиденья с закругленным передним краем; регулировку высоты поверхности сиденья в пределах 400 -550 мм и углам наклона вперед до 15 град, и назад до 5 град.; высоту опорной поверхности спинки 300±20 мм, ширину - не менее 380 мм и радиус кривизны горизонтальной плоскости - 400 мм; угол наклона спинки в вертикальной плоскости в пределах ±30 градусов; регулировку расстояния спинки от переднего края сиденья в пределах 260 - 400 мм; стационарные или съемные подлокотники длиной не менее 250 мм и шириной - 50 - 70 мм; регулировку подлокотников по высоте над сиденьем в пределах 230±30 мм и внутреннего расстояния между подлокотниками в пределах 350 - 500 мм.

Лица, работающие с ПЭВМ более 50% рабочего времени, должны проходить обязательные предварительные при поступлении на работу и периодические медицинские осмотры в установленном порядке. Женщины со времени установления беременности переводятся на работы, не связанные с использованием ПЭВМ, или для них ограничивается время работы с ПЭВМ (не более 3-х часов за рабочую смену) при условии соблюдения гигиенических требований, установленных санитарными правилами.

## 10.2 ИДЕНТИФИКАЦИЯ ОПАСНЫХ И ВРЕДНЫХ ПРОИЗВОДСТВЕННЫХ ФАКТОРОВ

При работе с ПЭВМ согласно ГОСТ 12.0.003-74\* «Опасные и вредные производственные факторы. Классификация» инженер подвергается воздействию многих факторов, которые можно разделить на 4 группы:

1. Химические – представляют собой вещества и их соединения, которые могут оказать неблагоприятное воздействие на пользователя.  
   На рабочем месте отсутствуют.
2. Биологические – связаны с повышенным содержанием в воздухе рабочей зоны патогенных микроорганизмов (вирусы, бактерии и т.д.)  
   Данная группа факторов так же отсутствует.
3. Психофизиологические – приводят к физическим и нервно-психическим перегрузкам. Они связаны с умственным перенапряжением, перенапряжением анализаторов, монотонностью труда, а также с длительным статическим напряжением мышц пользователя и динамическими перегрузками мышц пальцев рук при наборе большого количества знаков.  
    Разработчик занимается умственным видом деятельности, для нее характерна наибольшая степень напряжения внимания. Характерной при работе с ПЭВМ является такая физическая перегрузка, как длительное статическое напряжение мышц пользователя. Оно обусловлено продолжительным сидением в одной и той же рабочей позе, необходимостью постоянного наблюдения за экраном, набором большого количества знаков. При этом возникает и локальная динамическая перегрузка пальцев и кистей рук. Следовательно, психофизиологические факторы воздействуют на пользователя.
4. Физические – к ним относятся повышенные уровни переменных электростатических и электромагнитных полей (ЭМП), аномальный уровень освещённости рабочего места, повышенный уровень шума, повышенный уровень напряжения в электросети, аномальный микроклимат в помещении, возможность возникновения пожара и другие.  
   Электрический ток тоже имеет свое воздействие хотя бы потому, что оборудование питается от трехфазной сети переменного тока (380/220В, 50 Гц). В процессе эксплуатации или проведения профилактических работ человек может коснуться частей, находящихся под напряжением. Действие электрического тока на организм человека может вызвать травмы различной степени тяжести, и даже смертельный исход.

В помещениях, в которых работа с использованием ПЭВМ является основной и связана с нервно-эмоциональным напряжением, обеспечиваются оптимальные параметры микроклимата для категории работ 1а и 1б в соответствии с санитарно-эпидемиологическими нормативами микроклимата помещений. В помещениях проводится ежедневная влажная уборка и проветривание после каждого часа работы на ПЭВМ.

Особое внимание следует уделять повышенной запыленности воздуха, колебаниям температурного режима и освещенности.

Шумящее оборудование (печатающие устройства, серверы и т.п.), уровни шума которого превышают нормативные, должно размещаться вне помещений с ПЭВМ.

Уровень шума в помещениях при неработающей технике не превышает 40 дБА. При выполнении основной работы на ПЭВМ уровень шума на рабочем месте не превышает 50 дБА (СН 2.2.4/2.1.8.562-96).

Таблица 10.1 – Факторы, характеризующие условия труда

|  |  |  |
| --- | --- | --- |
| **Факторы, характеризующие условия труда** | **Единица измерения** | **Нормативная величина** |
| Требуемая площадь на одного человека | Кв.м. | 6,0 |
| Минимальная высота потолка | М | 3,2 |
| Скоростьдвижения воздуха:  холодный период  теплый период | м/сек | 0,1 (0,1)  0,1 (0,1-0,2) |
| Температура воздуха:  холодный период  теплый период | Град. С | 22-24(20-25)  23-25 (21-28) |
| Относительная влажность воздуха:  холодный период  теплый период | % | 40-60(15-75)  40-60(15-75) |
| Уровень шума | дБА | не более 50 |
| Освещение рабочей зоны | Лк | 300-500 лк |

**10.3 ВОЗДЕЙСТВИЕ ПРОИЗВОДСТВЕННЫХ ФАКТОРОВ**

Рабочие столы следует размещать таким образом, чтобы мониторы были ориентированы боковой стороной к световым проемам, чтобы естественный свет падал преимущественно слева. Рекомендуемые цвета поверхностей помещения приведены в таблице 10.2.

Таблица 10.2 – Рекомендуемые цвета поверхностей

|  |  |
| --- | --- |
| Ориентация окон помещения | Цвета поверхности |
| Юг | Стены – зеленовато-голубой или светло-голубой.  Пол – зелёный. |
| Север | Стены – оранжево-жёлтый или светло-оранжевый.  Пол - красновато-оранжевый. |
| Восток | Стены – желтовато-зелёный.  Пол – зелёный или красновато-оранжевый. |
| 3алад | Стены – светло-жёлтый или голубовато-зелёный  Пол – зелёный или красновато-оранжевый. |

Применяемые меры и средства электробезопасности должны обеспечивать безусловное выполнение требований ГОСТ 12.1.038-82\*, определяющего предельно допустимые значения напряжения прикосновения и протекающего через тело человека тока Iпд=0,3 мА при нормальном режиме.

Предельно допустимые уровни напряжения прикосновения и токов при аварийном режиме производственных электроустановок напряжением до 1000 В и частотой 50 Гц показаны в таблице 10.3.

Таблица 10.3 – Предельно допустимые уровни напряжения

|  |  |  |
| --- | --- | --- |
| **Продолжительность воздействия t, с** | **Uпд , В** | **Iпд ,мА** |
| 0.01…0.08 | 550 | 650 |
| 0.1-1.0 | 50/t | 50/t |
| 1 | 20 | 6 |

Искусственное освещение в помещениях для эксплуатации ПЭВМ должно осуществляться системой общего равномерного освещения. В административно-общественных помещениях, в случаях преимущественной работы с документами, следует применять системы комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения, предназначенные для освещения зоны расположения документов).

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Освещение не должно создавать бликов на поверхности экрана. Освещенность поверхности экрана не должна быть более 300 лк. Яркость светящихся поверхностей (окна, светильники и др.) находящихся в поле зрения, должна быть не более 200 кд/м2. Показатель дискомфорта в административно-общественных помещениях не более 40.

Яркость светильников общего освещения в зоне углов излучения от 50 до 90 градусов с вертикалью в продольной и поперечной плоскостях должна составлять не более 200 кд/м2, защитный угол светильников должен быть не менее 40 градусов. Светильники местного освещения должны иметь не просвечивающий отражатель с защитным углом не менее 40 градусов.

В качестве источников света при искусственном освещении следует применять преимущественно люминесцентные лампы типа ЛБ и компактные люминесцентные лампы (КЛЛ). При устройстве отраженного освещения в производственных и административно-общественных помещениях допускается применение металлогалогенных ламп. В светильниках местного освещения допускается применение ламп накаливания, в том числе галогенных.

Применение светильников без рассеивателей и экранирующих решеток не допускается.

Общее освещение при использовании люминесцентных светильников следует выполнять в виде сплошных или прерывистых линий светильников, расположенных сбоку от рабочих мест, параллельно линии зрения пользователя при рядном расположении видеодисплейных терминалов. При периметральном расположении компьютеров линии светильников должны располагаться локализовано над рабочим столом ближе к его переднему краю, обращенному к оператору.

Коэффициент запаса (К,) для осветительных установок общего освещения должен приниматься равным 1,4. Коэффициент пульсации не должен превышать 5%.

Для обеспечения нормируемых значений освещенности в помещениях для использования ПЭВМ следует проводить чистку стекол оконных рам и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп (СанПиН 2.2.2.542-96).

* 1. **МЕРОПРИЯТИЯ ПО РАЗРАБОТКЕ БЕЗОПАСНЫХ УСЛОВИЙ ТРУДА НА РАБОЧЕМ МЕСТЕ**

Все мероприятия по разработке безопасных условий труда на рабочем месте можно разделить на 4 составляющие:

* организационные: проведение бесед, лекций, инструктажа;
* эксплуатационные: правильная эксплуатация оборудования;
* режимные: например, запрет на курение в рабочем помещении;

технические: соблюдение правил и норм пожарной безопасности; максимально-возможное применение несгораемых материалов; применение материалов, которые при горении не выделяют вредных газов; наличие установки объемного газового тушения и устройства системы автоматической пожарной сигнализации, реагирующей на появление дыма; наличие первичных средств пожаротушения.

* 1. **ОБЕСПЕЧЕНИЕ ЭЛЕКТРОБЕЗОПАСНОСТИ НА РАБОЧЕМ МЕСТЕ**

Электробезопасность обеспечивается путем выполнения норм ГОСТ 12.1. 030. – 81. Опасное и вредное воздействие на людей электрического тока проявляется в виде электротравм и профессиональных заболеваний.

Электробезопасность в помещении обеспечивается техническими способами и средствами защиты: защитное зануление, защитное отключение, а также организационными и техническими мероприятиями. Основным организационным мероприятием является инструктаж и обучение безопасным методам труда, а также проверка знаний правил безопасности и инструкций в соответствии с занимаемой должностью применительно к выполняемой работе.

Степень опасного и вредного воздействия на человека электрического тока, электрической дуги и электромагнитных полей зависит от:

* рода и величины напряжения и тока
* частоты электрического тока
* пути тока через тело человека
* продолжительности воздействия на организм человека

Рассмотрим основные причины поражения человека электрическим током на рабочем месте.

* Прикосновение к металлическим нетоковедущим частям (корпусу, периферии компьютера), которые могут оказаться под напряжением в результате повреждения изоляции.
* Нерегламентированное использование электрических приборов.

В течении работы на корпусе компьютера накапливается статическое электричество. Кроме того, при неисправности каких-либо блоков компьютера корпус может оказаться под током, что может привести к электрическим травмам или электрическим ударам. Для устранения этого необходимо обеспечить подсоединение металлических корпусов оборудования к заземляющей жиле.

В рассматриваемом помещении на рабочем месте пользователя из всего оборудования металлическим является лишь корпус системного блока компьютера, но здесь используются системные блоки, в которых кроме рабочей изоляции предусмотрен элемент для заземления и провод с заземляющей жилой для присоединения к источнику питания.

Так как все токоведущие части ПЭВМ изолированы, то случайное прикосновение к токоведущим частям исключено.

Таким образом, данное помещение можно отнести к первой категории электробезопасности (помещения без повышенной опасности), так как температура и относительная влажность воздуха не превышают допустимых значений, отсутствует токопроводящая пыль и исключена возможность прикосновения к токоведущим частям ПЭВМ.

Также стоит помнить, что при проведении незапланированного и планового ремонта вычислительной техники выполняются следующие действия:

* Отключение компьютера от сети
* Проверка отсутствия напряжения
* После выполнения этих действий проводится ремонт неисправного оборудования.

Если ремонт проводится на токоведущих частях, находящихся под напряжением, то выполнение работы проводится не менее чем двумя лицами с применением электрозащитных средств.

* 1. **ОБЕСПЕЧЕНИЕ ПОЖАРНОЙ БЕЗОПАСНОСТИ НА РАБОЧЕМ МЕСТЕ**

Возникновение пожара на рабочем месте может быть следствием причин электрического и неэлектрического характера. К причинам электрического характера относятся следующие:

* перегрузка проводников;
* короткие замыкания.

Эти причины устраняются мероприятиями согласно ГОСТ 12.1.004-91:

* правильная эксплуатация оборудования;
* использование плавких предохранителей или автоматических аппаратов защиты;
* заземление всех металлических частей электроустановок, на которых могут скапливаться заряды.

Помещение, в котором будет располагаться рабочее место оператора ПЭВМ относится к категории В по взрывопожарной и пожарной опасности. Так как в нем присутствуют горючие и трудногорючие вещества и материалы в холодном состоянии, такие как кабельные линии, бумага, картон, мебель из горючих материалов.   
 Определим категорию помещения, где располагается рабочее место, для этого произведем расчет пожарной нагрузки по следующей формуле:

*,*

где Gi – количество i-го материала пожарной нагрузки, кг;

Q(p)Hi – низшая теплота сгорания i-го материала пожарной нагрузки, МДж\*кг-1;

S – площадь помещения.

Помещение имеет площадь равную 46,48 м². В нем находится 9 ПЭВМ весом 12 кг каждая, 10 деревянных стола весом по 24 кг, 9 стула весом 18 кг каждый. Перечисленные предметы занимают общую площадь 40,5 м².

Низшая теплота сгорания берется из таблицы по каждому веществу. Проводим расчет:

Q = (108\*47,14+162\*8,3+240\*13,8) / 40,5= 9775,8/40,5 = 241,38 МДж/ м².

Таблица 10.4 – Категория помещения в соответствии с пожарной нагрузкой

|  |  |
| --- | --- |
| Категория помещения | Удельная пожарная нагрузка g на участке, МДж/м2 |
| В1 | Более 2200 |
| В2 | 1401–2200 |
| В3 | 181–1400 |
| В4 | 1–180 |

Таким образом, рабочее помещение имеет категорию пожарной опасности В3.

В помещении имеется «План эвакуации людей при пожаре», регламентирующий действия сотрудников в случае возникновения очага возгорания и указывающий места расположения пожарной техники. В рассматриваемом помещении имеется один углекислотный огнетушитель - ОУ5 (ВВК3,5), расстояние от возможного очага пожара до него не превышает 30 м.

Также, для данного помещения (площадь менее 80 м2, высота потолков 3,5 м), достаточно необходимо и достаточно установить два точечных дымовых пожарных извещателя, на расстоянии 4,5 м от стены и 9 м между извещателями (в соответствии с НПБ 88 - 01).

* Обеспечен беспрепятственный доступ к аварийному выходу. Раз в полгода проводится инструктаж сотрудников по пожарной безопасности и технике безопасности при работе с ПЭВМ.

**10.7 ИНЖЕНЕРНЫЕ РАСЧЕТЫ**

Для расчета искусственного освещения применяем метод коэффициента использования светового потока.

Световой поток рассчитывается по формуле:

,

где - – нормированная минимальная освещенность (лк), для помещения с заданными параметрами норма освещенности для газоразрядных ламп 300 лк;

S – площадь освещаемого помещения, м ;

– коэффициент запаса, учитывающий запыленность светильников и

износ источников света в процессе эксплуатации;

Z – коэффициент неравномерности освещения, значение которого для

люминесцентных ламп.

N – число светильников в помещении;

n - число ламп в светильнике;

j - коэффициент использования излучаемого светильниками светового потока на расчетной плоскости, зависящей от КПД, кривой распределения силы света светильника, коэффициента отражения потолка (рn), стен (рс), высоты подвеса светильников и показателя помещения (i).

Произведем расчет для третьего разряда зрительных работ   
(подразряд – средний).

Высота расположения светильника над освещаемой поверхностью рассчитывается по формуле:

,

где H– высота помещения, м;

hс – расстояние от потолка до светильника, м;

hр – высота от пола до освещаемой поверхности, м.

h= 3,5-0,3-0,75=2,45 м.

Показатель помещения определяется по формуле:

,

где а – длина помещения, м;

b – ширина помещения, м.

Определим показатель помещения, рассмотренного выше и изображенного на Рисунке 1.

,

Зная индекс помещения, по таблицам находим коэффициент использования.



Необходимое количество светильников находим по формуле:





При эксплуатации светильники загрязняются, и освещенность снижается. Для учета снижения освещенности применяется коэффициент износа – kz. Пусть kz=1.4. Тогда:



По полученному результату выбираем тип ламп, которые будут применяться для освещения комнаты, – ЛД 80.

Электрическая мощность всей осветительной системы вычисляется по формуле:

,

где  – мощность одной лампы;

– число ламп;

– коэффициент запаса.



В помещении используются светильники с люминесцентными лампами ОД. Каждый светильник комплектуется одной лампой. Светильники размещены в специальных световых коробах двумя рядами по два в каждом (Рисунок 10.3).

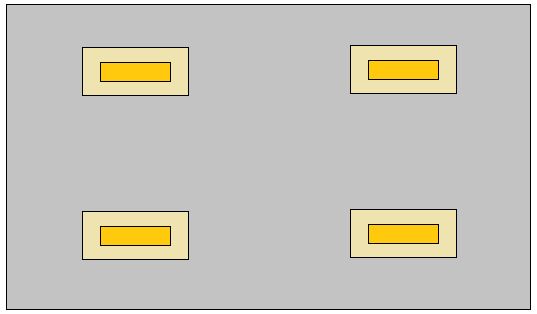


Рисунок 10.3 – Схема расположения светильников в помещении.

Проверка фактического уровня освещенности:



**10.8 ЭКОЛОГИЧНОСТЬ**

Рассмотрим опасные и вредные факторы, имеющие место в помещении с компьютером и их возможное влияние на окружающую среду. Вредное воздействие может оказывать энергопотребление. Это воздействие проявляется опосредованно, через вред, наносимый окружающей среде электро- и теплостанциями. Важным фактором является энергопотребление компьютера, минимальное энергопотребление современных компьютеров составляет около 150Вт, учитывая энергосберегающие технологии. А потребляемая мощность компьютера программиста составляет 450 Вт с учетом энергосберегающих технологий и использования ЖК-монитора с функцией «спящий режим».

Известно, что значительную роль в обеспечении экологической безопасности играет решение вопроса хранения, переработки и утилизации отходов. Все отходы можно разделить на две группы: твердые бытовые отходы (ТБО) и отходы производства. ТБО – это в основном бумага, пластмасса и стекло.

В настоящее время объективным сдерживающим фактором в развитии городской системы сбора, переработки отходов производства и потребления можно назвать такие задачи, как отсутствие нормативной базы, адекватной современным требованиям к состоянию экологической среды, транспортировки отходов, их хранения, переработки, сбыта переработанного сырья, захоронения и сжигания отходов. При сжигании выбросы должны удовлетворять СанПиН 2.1.6.983-00 «Гигиенические требования к обеспечению качества атмосферного воздуха населенных мест».  
 Разработка данного программного обеспечения в силу объективных причин не оказывает негативного воздействия на экологическую обстановку окружающей среды. Экологическая безопасность проекта является следствием выполнения всех норм и правил, предусмотренных нормативными документами.

**ЗАКЛЮЧЕНИЕ**

В данном дипломном проекте было разработано программное обеспечение для прогнозирования показателей портфеля автострахования. Основное назначение ПО – повышение точности прогнозирования показателей портфеля посредством разработки программного обеспечения, анализирующего структуру портфеля автострахования с помощью алгоритмов анализа данных.

Основным достоинством разработанного ПО является то, что оно просто в использовании, но в то же время предоставляет большие функциональные возможности и гибкость анализа.

В ходе дипломного проектирования было выполнено следующее:

* разработан алгоритм загрузки информации из CSV файлов с информацией о договорах за прошлый период;
* разработан пользовательский интерфейс, способствующий быстрому обучению работы с программой и снижения количества ошибок;
* выполнена программная реализация на языке C#, с использованием Microsoft Visual Studio 2013 Ultimate, технологии .NET и графической подсистемы Windows Presentation Foundation.

Разработанный программный продукт позволяет:

* проводить анализ договоров страхования;
* осуществлять прогноз показателей портфеля на основе этого анализа.

Также в ходе дипломного проектирования была проведена оценка экономической эффективности программного продукта, проведен анализ безопасности и экологичности проекта для выявления и устранения вредных факторов, действующих на человека при работе на ПЭВМ.

Разработанное ПО имеет большие возможности для расширения функциональности и быстрого добавления новых видов анализа, исходя из потребностей пользователей. В перспективе планируется усовершенствовать программное обеспечение, добавив возможность автоматического формирования исходных данных, а также сделать процессы анализа и прогнозирования независимыми от формы представления данных, что придаст программному обеспечению универсальности и позволит использовать его, помимо рынка автострахования, на других рынках страхования. Также планируется разработка Web-версии продукта.

Таким образом, в настоящем дипломном проекте была достигнута цель и решены поставленные задачи. Разработанное программное обеспечение соответствует требованиям задания к дипломному проекту.

**БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. ГОСТ 12.0.003-74 «ССБТ. Опасные и вредные производственные факторы»
2. ГОСТ 12.1.004-91 «Пожарная безопасность. Общие требования»
3. ГОСТ 12.1.019-79 «ССБТ. Электробезопасность. Общие требования»
4. ГОСТ 12.1.038-82 «ССБТ. Электробезопасность. Предельно допустимые уровни напряжений прикосновения и токов»
5. ГН 2.2.5.1313-03 «Предельно допустимые концентрации (ПДК) вредных веществ в воздухе рабочей зоны»
6. ГОСТ Р МЭК 60065-2005 «Аудио-, видео-, и аналогичная электронная аппаратура. Требования безопасности»
7. Грэди Б. Язык UML. Руководство пользователя. М.: ДМК – Пресс, 2007. 494с**.**
8. Гультяев А., Машин В. Проектирование и дизайн пользовательского интерфейса. Корона-принт, 2010.
9. Дегтярева О.И. Страховое дело. М.: ЮНИТИ, 2004.
10. Карпиков Е.И., Рубин Ю.В., Солдаткин В.И. Рынок автострахования М.: СОМИНТЭК, 2006.
11. Кнорринг Г.М. Осветительные установки. Л.: Энергоиздат. Лениигр. отд, 1981. 288 с.
12. Косаренко Н.Н. Правовое обеспечение публичных интересов в сфере страхования. М.: БИЗНЕС, 2010. – 301 с.
13. Мак-Дональд, М. WPF: Windows Presentation Foundation в .NET 4.0 с примерами на C# 2010 для профессионалов. – М.: ВИЛЬЯМС, 2011. – 1024 с.
14. Рутковская, Д. Алгоритмы анализа данных/ Д.Рутковская, М. Пичиньский, Л. Рутковский. – М.: Горячая линия-Телеком, 2008. – 351 с.
15. СанПиН 2.2.2*/*2.4.1340-03 «Гигиенические требования к ПЭВМ и организации работы»
16. СанПиН 2.2.4.548-96 «Гигиенические требования к микроклимату производственных помещений»
17. Хайк, А. Алгоритмы и методы кластеризации. – М.: ВИЛЬЯМС, 2005. – 804 с.
18. Шилдт, Г. C# 4.0 полное руководство. – М.: ВИЛЬЯМС, 2011. – 1056 с.
19. Экономические вопросы проектирования. Расчеты экономической эффективности в дипломных проектах: Методические указания к дипломному проектированию / Рязан. гос. радиотехн. акад.; Сост.: Мисник Е.В., Прудников Ю.Н. – Рязань, 2000.
20. <http://www.artlebedev.ru/tools/colors/>
21. [http:// www.msdn.microsoft.com/](http://msdn.microsoft.com/)

**ПРИЛОЖЕНИЕ**

/// Тема дипломного проекта: Разработка программного обеспечения

///для анализа структуры портфеля автострахования физических лиц

/// Дипломник: Башева И.А.

/// Специальность 080801, группа 944.

/// Научный руководитель: канд. техн. наук, ст. преподаватель Коротаев А.Н.

/// Средство разработки: Microsoft Visual Studio 2013 Ultimate

/// Дата разработки: 01.05.2014.

/// Программное обеспечение

**Код разметки App.xaml**

<Application x:Class="InsPort.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

StartupUri="View\MainWindow.xaml">

<Application.Resources>

</Application.Resources>

</Application>

**Класс App.xaml.cs**

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Threading.Tasks;

using System.Windows;

namespace InsPort

{

/// <summary>

/// Логика взаимодействия для App.xaml

/// </summary>

public partial class App : Application

{

public App()

{

}

}

}

**Класс SettingsModel.cs**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Linq;

using System.Text;

namespace InsPort.Model

{

class SettingsModel : INotifyPropertyChanged //класс Model для окна настроек SettingsWindow

{

//вспомогательное событие для отслежевания обновления данных через Code Behind

#region Implement INotyfyPropertyChanged members

public event PropertyChangedEventHandler PropertyChanged;

protected virtual void OnPropertyChanged(string propertyName)

{

if (PropertyChanged != null)

{

PropertyChanged(this, new PropertyChangedEventArgs(propertyName));

}

}

#endregion

//закрытые поля класса, обеспечивающие двунаправленное обновление данных через Code Behind

#region Fields

private int \_kmin; //поле, сохраняющее начальное количество кластеров разбиения для алгоритма k-средних

private int \_kmax; //поле, сохраняющее конечное количество кластеров разбиения для алгоритма k-средних

private int \_nstep; //поле, сохраняющее максимальное количество итерация для алгоритма k-средних

private double \_r; //поле, сохраняющее значение коэффициента отталкивания алгоритма CLOPE

#endregion

//свойства класса

#region Properties

public int kmin //свойство, сохраняющее начальное количество кластеров разбиения для алгоритма k-средних

{

get { return \_kmin; }

set

{

if (\_kmin != value)

{

\_kmin = value;

OnPropertyChanged("kmin");

}

}

}

public int kmax //свойство, сохраняющее конечное количество кластеров разбиения для алгоритма k-средних

{

get { return \_kmax; }

set

{

if (\_kmax != value)

{

\_kmax = value;

OnPropertyChanged("kmax");

}

}

}

public int nstep //поле, сохраняющее максимальное количество итерация для алгоритма k-средних

{

get { return \_nstep; }

set

{

if (\_nstep!= value)

{

\_nstep = value;

OnPropertyChanged("nstep");

}

}

}

public double r //свойство, сохраняющее значение коэффициента отталкивания алгоритма CLOPE

{

get { return \_r; }

set

{

if (\_r != value)

{

\_r = value;

OnPropertyChanged("r");

}

}

}

#endregion

}

}

**Класс SettingsViewModel.cs**

using InsPort.Algorithms;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows.Input;

using Xceed.Wpf.Toolkit;

namespace InsPort.ViewModel

{

class SettingsViewModel : Model.SettingsModel //класс ViewModel для окна настроек SettingsWindow

{

//конструктор класса SettingsViewModel, связывающий свойства класса с свойствами класса SettingsModel

#region Constructor

public SettingsViewModel(int kmin, int kmax, int nstep, double r)

{

ClickCommand = new Command(arg => Load());

Settings = new Model.SettingsModel()

{

kmin = kmin,

kmax = kmax,

nstep = nstep,

r = r,

};

}

#endregion

#region Properties

public Model.SettingsModel Settings { get; set; }

#endregion

#region Commands

public ICommand ClickCommand { get; set; }

#endregion

#region Methods

private void Load()

{

SettingsViewModel Settings;

Settings = new SettingsViewModel(InsPort.Settings.kmin, InsPort.Settings.Max, InsPort.Settings.nstep, InsPort.Settings.r);

}

#endregion

}

}

**Код класса CSVReader.cs**

using System;

using System.Collections.Generic;

using System.Data;

using System.IO;

using System.Linq;

using System.Text;

namespace InsPort.Algorithms

{

/// <summary>

/// Класс считывания данных из файла с расширением .csv

/// </summary>

public class CSVReader : IDisposable

{

public const string NEWLINE = "\r\n";

/// <summary>

/// Считывание всех данных из файла

/// </summary>

private BinaryReader reader;

/// <summary>

/// Количество строк для считывания при формировании data table

/// </summary>

public int ScanRows = 0;

#region Constructors

/// <summary>

/// Чтение данных в формате csv из файла

/// </summary>

/// <param name="filename">Имя файла</param>

public CSVReader(FileInfo csvFileInfo)

{

if (csvFileInfo == null)

throw new ArgumentNullException("Отсутствует информация о файле!");

this.reader = new BinaryReader(File.OpenRead(csvFileInfo.FullName));

}

/// <summary>

/// Чтение данных в формате csv из строки

/// </summary>

/// <param name="csvData">String containing CSV data</param>

public CSVReader(string csvData)

{

if (csvData == null)

throw new ArgumentNullException("Пустая строка!");

this.reader = new BinaryReader(new MemoryStream(System.Text.Encoding.UTF8.GetBytes(csvData)));

}

/// <summary>

/// Чтение данных из text reader

/// </summary>

/// <param name="reader">имя text reader</param>

public CSVReader(TextReader reader)

{

if (reader == null)

throw new ArgumentNullException("Отсутствует text reader!");

this.reader = new BinaryReader(new MemoryStream(System.Text.Encoding.UTF8.GetBytes(reader.ReadToEnd())));

}

#endregion

string currentLine = "";

/// <summary>

/// Чтение следубщей строки из файла

/// </summary>

/// <returns>Лист считанных объектов</returns>

public List<object> ReadRow()

{

if (reader.BaseStream.Position >= reader.BaseStream.Length)

return null;

StringBuilder builder = new StringBuilder();

// Считывание следующей строки

while ((reader.BaseStream.Position < reader.BaseStream.Length) && (!builder.ToString().EndsWith(NEWLINE)))

{

char c = reader.ReadChar();

builder.Append(c);

}

currentLine = builder.ToString();

if (currentLine.EndsWith(NEWLINE))

currentLine = currentLine.Remove(currentLine.IndexOf(NEWLINE), NEWLINE.Length);

// Создание листа строк

List<object> objects = new List<object>();

while (currentLine != "")

objects.Add(ReadNextObject());

return objects;

}

/// <summary>

/// Чтение следующего объекта из правильно сформированной строки

/// </summary>

/// <returns>Правильно сформированная строка</returns>

private object ReadNextObject()

{

if (currentLine == null)

return null;

//Проверка того, что следующая строка с кавычками

bool quoted = false;

if (currentLine.StartsWith("\""))

quoted = true;

//Нахождение конца считываемой строки

string nextObjectString = "";

int i = 0;

int len = currentLine.Length;

bool foundEnd = false;

while (!foundEnd && i <= len)

{

// Проверка конца строки

if ((!quoted && i == len) // строка без кавычек должна заввершаться символом точка с запятой

|| (!quoted && currentLine.Substring(i, 1) == ";")

// строка в кавычках должна завершаться точкой с запятой или символом конца строки

|| (quoted && i == len - 1 && currentLine.EndsWith("\""))

|| (quoted && currentLine.Substring(i, 2) == "\";"))

foundEnd = true;

else

i++;

}

if (quoted)

{

if (i > len || !currentLine.Substring(i, 1).StartsWith("\""))

throw new FormatException("Invalid CSV format: " + currentLine.Substring(0, i));

i++;

}

nextObjectString = currentLine.Substring(0, i).Replace("\"\"", "\"");

if (i < len)

currentLine = currentLine.Substring(i + 1);

else

currentLine = "";

if (quoted)

{

if (nextObjectString.StartsWith("\""))

nextObjectString = nextObjectString.Substring(1);

if (nextObjectString.EndsWith("\""))

nextObjectString = nextObjectString.Substring(0, nextObjectString.Length - 1);

return nextObjectString;

}

else

{

object convertedValue;

StringConverter.ConvertString(nextObjectString, out convertedValue);

return convertedValue;

}

}

/// <summary>

/// Считывание данных методом ReadRow()

/// </summary>

/// <param name="headerRow">Истина, если первая строка содержит заголовки</param>

/// <returns>Объект data table</returns>

public DataTable CreateDataTable(bool headerRow)

{

//Чтение данных в формате csv

List<List<object>> rows = new List<List<object>>();

List<object> readRow = null;

while ((readRow = ReadRow()) != null)

rows.Add(readRow);

//Лист имен заголовков

List<Type> columnTypes = new List<Type>();

List<string> columnNames = new List<string>();

//Чтение заголовков столбцов

if (headerRow)

foreach (object name in rows[0])

columnNames.Add(name.ToString());

//Считывание имен типов столбцов

bool headerRead = false;

foreach (List<object> row in rows)

if (headerRead || !headerRow)

for (int i = 0; i < row.Count; i++)

//Запись типов в лист типов

if (columnTypes.Count < i + 1)

columnTypes.Add(row[i].GetType());

else

columnTypes[i] = StringConverter.FindCommonType(columnTypes[i], row[i].GetType());

else

headerRead = true;

//Создание таблицы и столбцов

DataTable table = new DataTable();

for (int i = 0; i < columnTypes.Count; i++)

{

table.Columns.Add();

table.Columns[i].DataType = columnTypes[i];

if (i < columnNames.Count)

table.Columns[i].ColumnName = columnNames[i];

}

//Добавление данных

headerRead = false;

foreach (List<object> row in rows)

if (headerRead || !headerRow)

{

DataRow dataRow = table.NewRow();

for (int i = 0; i < row.Count; i++)

dataRow[i] = row[i];

table.Rows.Add(dataRow);

}

else

headerRead = true;

return table;

}

/// <summary>

///Считывание данных в data table

/// </summary>

/// <param name="filename">Имя csv файла</param>

/// <param name="headerRow">Истина, если первая строка сожержит имена заголовков столбцов</param>

/// <param name="scanRows">Количество строк для сканирования</param>

/// <returns>Объект data table</returns>

public static DataTable ReadCSVFile(string filename, bool headerRow, int scanRows)

{

using (CSVReader reader = new CSVReader(new FileInfo(filename)))

{

reader.ScanRows = scanRows;

return reader.CreateDataTable(headerRow);

}

}

/// <summary>

///Считывание данных из csv файла

/// </summary>

/// <param name="filename">Имя файла</param>

/// <param name="headerRow">Истина если первая строка содержит имена заголовков столбцов</param>

/// <returns>Объект data table</returns>

public static DataTable ReadCSVFile(string filename, bool headerRow)

{

using (CSVReader reader = new CSVReader(new FileInfo(filename)))

return reader.CreateDataTable(headerRow);

}

#region IDisposable Members

public void Dispose()

{

if (reader != null)

{

try

{

reader.Close();

}

catch { }

}

}

#endregion

}

}

**Код класса ConvertString.cs**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace InsPort.Algorithms

{

/// <summary>

/// Класс преобразования строк к типизированным значениям

/// </summary>

public static class StringConverter

{

public static Type ConvertString(string value, out object convertedValue)

{

// определение типа

byte byteResult;

if (byte.TryParse(value, out byteResult))

{

convertedValue = byteResult;

return typeof(byte);

}

short shortResult;

if (short.TryParse(value, out shortResult))

{

convertedValue = shortResult;

return typeof(short);

}

int intResult;

if (int.TryParse(value, out intResult))

{

convertedValue = intResult;

return typeof(int);

}

long longResult;

if (long.TryParse(value, out longResult))

{

convertedValue = longResult;

return typeof(long);

}

ulong ulongResult;

if (ulong.TryParse(value, out ulongResult))

{

convertedValue = ulongResult;

return typeof(ulong);

}

float floatResult;

if (float.TryParse(value, out floatResult))

{

convertedValue = floatResult;

return typeof(float);

}

double doubleResult;

if (double.TryParse(value, out doubleResult))

{

convertedValue = doubleResult;

return typeof(double);

}

decimal decimalResult;

if (decimal.TryParse(value, out decimalResult))

{

convertedValue = decimalResult;

return typeof(decimal);

}

bool boolResult;

if (bool.TryParse(value, out boolResult))

{

convertedValue = boolResult;

return typeof(bool);

}

char charResult;

if (char.TryParse(value, out charResult))

{

convertedValue = charResult;

return typeof(char);

}

convertedValue = value;

return typeof(string);

}

/// <summary>

/// Соединение нескольких типов

/// </summary>

/// <param name="typeA">Первый тип</param>

/// <param name="typeB">Второй тип</param>

/// <returns>Возвращаемое соединение типов</returns>

public static Type FindCommonType(Type typeA, Type typeB)

{

BuildTypeMap();

if (!typeMap.ContainsKey(typeA))

return typeof(string);

if (!typeMap[typeA].ContainsKey(typeB))

return typeof(string);

return typeMap[typeA][typeB];

}

// Словарь типов

private static Dictionary<Type, Dictionary<Type, Type>> typeMap = null;

private static object locker = new object();

/// <summary>

/// Составление карты типов

/// </summary>

private static void BuildTypeMap()

{

lock (locker)

{

if (typeMap == null)

{

typeMap = new Dictionary<Type, Dictionary<Type, Type>>()

{

// Объединение byte

{typeof(byte), new Dictionary<Type, Type>() {

{ typeof(byte), typeof(byte) },

{ typeof(short), typeof(short) },

{ typeof(int), typeof(int) },

{ typeof(long), typeof(long) },

{ typeof(ulong), typeof(ulong) },

{ typeof(float), typeof(float) },

{ typeof(double), typeof(double) },

{ typeof(decimal), typeof(decimal) },

{ typeof(bool), typeof(string) },

{ typeof(char), typeof(string) },

{ typeof(string), typeof(string) },

}},

// Объединение short

{typeof(short), new Dictionary<Type, Type>() {

{ typeof(byte), typeof(short) },

{ typeof(short), typeof(short) },

{ typeof(int), typeof(int) },

{ typeof(long), typeof(long) },

{ typeof(ulong), typeof(ulong) },

{ typeof(float), typeof(float) },

{ typeof(double), typeof(double) },

{ typeof(decimal), typeof(decimal) },

{ typeof(bool), typeof(string) },

{ typeof(char), typeof(string) },

{ typeof(string), typeof(string) },

}},

// Объединение int

{typeof(int), new Dictionary<Type, Type>() {

{ typeof(byte), typeof(int) },

{ typeof(short), typeof(int) },

{ typeof(int), typeof(int) },

{ typeof(long), typeof(long) },

{ typeof(ulong), typeof(ulong) },

{ typeof(float), typeof(float) },

{ typeof(double), typeof(double) },

{ typeof(decimal), typeof(decimal) },

{ typeof(bool), typeof(string) },

{ typeof(char), typeof(string) },

{ typeof(string), typeof(string) },

}},

// Объединение long

{typeof(long), new Dictionary<Type, Type>() {

{ typeof(byte), typeof(long) },

{ typeof(short), typeof(long) },

{ typeof(int), typeof(long) },

{ typeof(long), typeof(long) },

{ typeof(ulong), typeof(ulong) },

{ typeof(float), typeof(float) },

{ typeof(double), typeof(double) },

{ typeof(decimal), typeof(decimal) },

{ typeof(bool), typeof(string) },

{ typeof(char), typeof(string) },

{ typeof(string), typeof(string) },

}},

// Объединение ulong

{typeof(ulong), new Dictionary<Type, Type>() {

{ typeof(byte), typeof(ulong) },

{ typeof(short), typeof(ulong) },

{ typeof(int), typeof(ulong) },

{ typeof(long), typeof(ulong) },

{ typeof(ulong), typeof(ulong) },

{ typeof(float), typeof(float) },

{ typeof(double), typeof(double) },

{ typeof(decimal), typeof(decimal) },

{ typeof(bool), typeof(string) },

{ typeof(char), typeof(string) },

{ typeof(string), typeof(string) },

}},

// Объединение float

{typeof(float), new Dictionary<Type, Type>() {

{ typeof(byte), typeof(float) },

{ typeof(short), typeof(float) },

{ typeof(int), typeof(float) },

{ typeof(long), typeof(float) },

{ typeof(ulong), typeof(float) },

{ typeof(float), typeof(float) },

{ typeof(double), typeof(double) },

{ typeof(decimal), typeof(decimal) },

{ typeof(bool), typeof(string) },

{ typeof(char), typeof(string) },

{ typeof(string), typeof(string) },

}},

// Объединение double

{typeof(double), new Dictionary<Type, Type>() {

{ typeof(byte), typeof(double) },

{ typeof(short), typeof(double) },

{ typeof(int), typeof(double) },

{ typeof(long), typeof(double) },

{ typeof(ulong), typeof(double) },

{ typeof(float), typeof(double) },

{ typeof(double), typeof(double) },

{ typeof(decimal), typeof(decimal) },

{ typeof(bool), typeof(string) },

{ typeof(char), typeof(string) },

{ typeof(string), typeof(string) },

}},

// Объединение decimal

{typeof(decimal), new Dictionary<Type, Type>() {

{ typeof(byte), typeof(decimal) },

{ typeof(short), typeof(decimal) },

{ typeof(int), typeof(decimal) },

{ typeof(long), typeof(decimal) },

{ typeof(ulong), typeof(decimal) },

{ typeof(float), typeof(decimal) },

{ typeof(double), typeof(decimal) },

{ typeof(decimal), typeof(decimal) },

{ typeof(bool), typeof(string) },

{ typeof(char), typeof(string) },

{ typeof(string), typeof(string) },

}},

// Объединение bool

{typeof(bool), new Dictionary<Type, Type>() {

{ typeof(byte), typeof(string) },

{ typeof(short), typeof(string) },

{ typeof(int), typeof(string) },

{ typeof(long), typeof(string) },

{ typeof(ulong), typeof(string) },

{ typeof(float), typeof(string) },

{ typeof(double), typeof(string) },

{ typeof(decimal), typeof(string) },

{ typeof(bool), typeof(bool) },

{ typeof(char), typeof(string) },

{ typeof(string), typeof(string) },

}},

// Объединение char

{typeof(char), new Dictionary<Type, Type>() {

{ typeof(byte), typeof(string) },

{ typeof(short), typeof(string) },

{ typeof(int), typeof(string) },

{ typeof(long), typeof(string) },

{ typeof(ulong), typeof(string) },

{ typeof(float), typeof(string) },

{ typeof(double), typeof(string) },

{ typeof(decimal), typeof(string) },

{ typeof(bool), typeof(string) },

{ typeof(char), typeof(char) },

{ typeof(string), typeof(string) },

}},

// Объединение string

{typeof(string), new Dictionary<Type, Type>() {

{ typeof(byte), typeof(string) },

{ typeof(short), typeof(string) },

{ typeof(int), typeof(string) },

{ typeof(long), typeof(string) },

{ typeof(ulong), typeof(string) },

{ typeof(float), typeof(string) },

{ typeof(double), typeof(string) },

{ typeof(decimal), typeof(string) },

{ typeof(bool), typeof(string) },

{ typeof(char), typeof(string) },

{ typeof(string), typeof(string) },

}},

};

}

}

}

}

}