

ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

на \_\_\_\_\_ дипломный проект  
(наименование вида работы)  
Хен Софья Константиновна  
\_\_\_\_\_ (Ф.И.О. обучающегося)  
5B070400 – Вычислительная Техника и Программное Обеспечение  
(шифр и наименование специальности)

На тему: «Разработка системы для автоматизации тестирования в коммунальной сфере»

Оценка работы

Актуальность темы дипломного проекта заключается в том, что ошибки в отчетах могут нанести огромный ущерб компании, а также и заказчикам. Автоматизация тестирования помогает исключить ошибки, совершенные из-за человеческого фактора, невнимательности, незнания предметной области. Результаты тестирования направлены на то, чтобы уведомить разработчиков и ответственных за отчеты людей о допущенном просчете. Дипломный проект состоит из 3-х глав, включая аннотацию на трех языках, введение, заключение и приложение.

В первой главе сделан анализ предметной области, деятельности компании, обоснование выбора программного обеспечения. Также раскрыты цели и задания для разработки проекта, составлено техническое задание.

Во второй главе представлены необходимые uml- диаграммы, er-диаграммы, в которых показано, как должна работать программа, какие данные будут использоваться и где будут храниться.


В третьей главе показан интерфейс программы и скриншоты со всеми возможными действиями в программе, так же описание этих действий согласно техническому заданию.

В целом данный дипломный проект выполнен с учетом всех требований, предъявляемых к дипломному проекту по специальности 5B070400 – «Вычислительная Техника и Программное Обеспечение», студент Хен С. рекомендован к защите дипломного проекта и заслуживает присвоения академической степени «бакалавра» по специальности 5B070400 – «Вычислительная Техника и Программное Обеспечение».

Научный руководитель

лектор

(должность, уч. степень, звание)

 Омарова Г. А.

(подпись)

« 16 » \_\_\_\_\_ 2019г.

## Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Хен Софья

**Название:** Хен Софья

**Координатор:** Гульнара Омарова

**Коэффициент подобия 1:**17

**Коэффициент подобия 2:**0,7

**Тревога:**1

**После анализа Отчета подобия констатирую следующее:**

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

.....  
.....  
.....  
.....  
.....  
.....

16.05.19

Дата



Подпись Научного руководителя

Хен С.К.

Корпоративная система регистрации посетителей

**ПОСНИТЕЛЪНА ЗАПИСКА**

к дипломному проекту

Специальность 5В070400 – Вычислительная техника и программное  
обеспечение



**ДОПУЩЕН К ЗАЩИТЕ**

Заведующий кафедрой ИИ  
канд. техн. наук, доцент,

ассистент-профессор

Р. Юнусов  
" 14 " 05  
2019 г.

**ПОСНИТЕЛЪНА ЗАПИСКА**

к дипломному проекту

На тему: "Корпоративная система регистрации посетителей"

по специальности 5B070400 – Вычислительная техника и программное  
обеспечение

Выполнил

Хен С.К.

Научный руководитель

лектор

Г.А. Омарова  
" 16 " 05  
2019 г.

**ҰТВЕРЖАЮ**

Заведующий кафедрой ИИ

канд. техн. наук, доцент,

ассистент-профессор

Р. Юнусов

"17" "05"  
2019 г.

**ЗАЯВЛЕНИЕ**

**на выполнение дипломного проекта**

Обучающемуся Хен Софье Константиновне

Тема: Разработка системы для автоматизации тестирования в

коммунальной сфере

Утверждена приказом ректора по академической работе № 1162 - 6

от "16" октября 2018 г.

"17" мая 2019 г.

Исходные данные к дипломному проекту: техническая документация по

применению технологий, технологическое задание, описание ВД для хранения

информации в виде ER-диаграммы.

Перечень подлежащих разработке в дипломном проекте вопросов:

а) Разработка новой базы данных для хранения результатов проведенного

тестирования;

б) Написание процедуры для формирования отчетов;

в) Создание параметризированной процедуры для автоматизации

тестирования;

г) проектирование и разработка пользовательского интерфейса;

д) разработка, отладка, тестирование программного комплекса.

Перечень графического материала (с точным указанием обязательных

чертежей): представлены 14 слайдов презентации.

Рекомендуемая основная литература: из наименований.



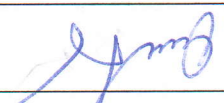

## ГРАФИК

подготовки дипломного проекта

Сроки представления научному руководителю и консультантам	Наименование разделов, перечень разрабатываемых вопросов	Примечание
	1. Анализ предметной области, разработка технического задания на проектирование программного комплекса	—
09.01.2019	2. Реализация классовой структуры модели базы	—
11.01.2019	3. Создание процедур для формирования таблиц с результатами отчета, создание процедуры для проверки данных между отчетами	—
01.02.2019	4. Отладка процедур, проверка на различных объемах данных	—
05.02.2019	5. Разработка дизайна интерфейса, разработка самого интерфейса с помощью с# и windows Form	—
23.04.2019	6. Написание пояснительной записки к дипломному проекту	—

### Подписи

консультантов и нормоконтролера на законченный дипломный проект  
с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Программное обеспечение	Өмірбекова З.М., ассистент	01.08.19	
Нормоконтролер	Г.А. Омарова, лектор		

Научный руководитель

Г.А. Омарова

Задание принял к исполнению обучающийся

С.К. Хен

Дата

" 11 " августа 2019 г.

## АҢДАТПА

Осы бағдарламалық қамтамасыз ету бөлімінің үшін арналған бағдарламалық қамтамасыз етуді әзірлеу және қолдау бөлімінің сапасы мен ақпараттық жүйелерді тестілеу басқармасының бөлім "жобасы бойынша жылу энергиясы". Шешім жасау мүмкіндігін болдырмауға мүмкіндік береді, жана деректерді тексеру кезінде қателер салдарынан процесін жылдамдату үшін ұсынылып отырған Есептілік нысандары адам факторын сводимости қалыптастырылған есеп. Қосымша бағдарламаларды орнату және байланыс сервері Дерекқорлармен жұмыс істеу үшін талап етілмейді, тек қоса бере отырып рұқсаты болуы керек.

Тестілеу жазбада автоматтандыру мәселелері қаралады, қажет болған себептері ол құрылған. Бұл жоба бағдарламалау тілінің көмегімен әзірленді және графикалық интерфейстің реляциялық дерекқордың C #, Windows Forms Microsoft SQL Server. Microsoft Visual Studio Professional бағдарламалау ортасы ретінде Psion 2015.



## АННОТАЦИЯ

Данное программное обеспечение предназначено для отдела разработки и поддержки программного обеспечения, отдела тестирования качества и отдела управления информационными системами по проекту «Теплоэнергия». Предлагаемое решение позволяет исключить возможность совершения ошибок при проверке новых отчетных форм из-за человеческого фактора, для ускорения процесса сводимости данных в сформированных отчетах. Не требуется установка дополнительных программ для работы с приложением, только надо иметь доступ на сервер и связь с базами данных.

В пояснительной записке рассматриваются вопросы автоматизации тестирования, причины необходимости его создания. Данный проект был разработан с помощью языка программирования C#, графического интерфейса Windows Forms и реляционной базы данных Microsoft SQL Server. В качестве среды программирования был использован Microsoft Visual Studio Professional 2015.

## ANNOTATION

This software is intended for department of development and support of the software, department of testing of quality and department of management of information systems of the Heat power project. The proposed solution allows to exclude a possibility of commission of mistakes when checking new forms of account because of a human factor, for acceleration of process of reducibility of data in the created reports. Installation of additional programs for work with the application is not required, it is only necessary to have access to the server and communication with databases.

In the explanatory note questions of automation of testing, the reason of need of its creation are considered. This project was developed by means of the C# programming language, the graphic Windows Forms interface and the relational Microsoft SQL Server database. As a programming environment Microsoft Visual Studio Professional 2015 was used.

## СОДЕРЖАНИЕ

	Введение	9
1	Теоретическая часть	11
1.1	Общая информация о компании	11
1.2	Термины и аббревиатуры	11
1.3	Чем занимается компания	11
1.4	Анализ предметной области	13
1.5	Требования к разработке	14
1.5.1	Общие сведения, назначение и цели разработки	14
1.5.2	Характеристика отчетов и справочников	15
1.5.3	Состав и сроки работ по созданию системы	15
1.5.4	Порядок контроля и приемки системы	15
1.5.5	Требования к безопасности	16
1.6	Технологии, используемые для создания программного обеспечения для автоматизации тестирования	16
2	Практическая часть	18
2.1	UML-диаграммы	18
3	Проектная часть	23
3.1	Интерфейс приложения	23
	Заключение	30
	Список использованной литературы	31
	Сноски	33
	Приложение А. Текст программы	34



## ВВЕДЕНИЕ

В современном мире и обществе автоматизация деятельности является неотъемлемой частью абсолютно любого предприятия, правда не всегда и не везде реализована. Контроль над разными процессами с помощью автоматизированной системы компьютера помогает добиться более высокой производительности труда и сберечь достаточно времени и средств. Высококачественная автоматизация технологических процессов довольно сильно облегчает и сокращает работу некоторых кадров предприятия и производства в целом.

Какие же предпосылки для создания автоматизации? Естественно, это большие потери рабочего времени, материальных, а так же трудовых ресурсов на ведение, проверку, контроль документации, систематизация и поддержание данных в достоверном и стабильном состоянии; неоправданно большое количество ошибок при выполнении выборок из каких-то необходимых данных и проверке данных по различным отчетам, как новым, так и давно созданным. Исправление же ошибок и их последствий – очень затратная составляющая статья расходов, средством сокращения которых является автоматическое тестирование.

Целью этой выпускной квалификационной практической работы является разработка автоматизированной информационной системы, обеспечивающей хранение информации о проведенных тестах на правильность данных в различных отчетах, проверке на расходимость сведений в этих отчетах, в каком из отчетов была допущена эта погрешность, в чем именно различаются эти данные и на сколько.

Самым главным компонентом разрабатываемого мной программного средства является реляционная база данных Microsoft SQL Server . Для реализации системы была создана логическая и физическая модель базы данных.

Для разработки программного средства необходимо решить задачи анализа, и, конечно же, синтеза.

Определим задачи анализа:

1. Изучение предметной области;
2. Выбор из средств проектирования базы данных;
3. Анализ аналогов программных системы;
4. Выбор средств и программ для проектирования программного средства;
5. Выбор необходимых видов отчетов для сравнения и данных в них, определение, где содержатся эти данные и как ни связаны между собой;

Синтез в себе содержит:

1. Создание логической и физической модели базы данных, ее проектирование;

2. Создание интерфейса для автоматизированной системы;
3. Воплощение алгоритмов данной программной системы;
4. Реализация целостности и безопасности данных;
5. Отладка и тестирование процедур, а после и самого программного средства;
6. Ведение соответствующей документации;
7. Сопровождение созданной автоматизированной системы.

Созданное мной программное средство необходимо для автоматизации работы для авто тестирования на наличие ошибок при сравнение данных в отчетах.

## 1. Теоретическая часть

### 1.1. Общая информация о компании

АЛСЕКО (Almaty Service Computing) - IT-компания, работающая в Республике Казахстан уже 20 лет. Больше всего она предоставлением IT услуг поставщикам коммунальных услуг, а именно – печати квитанций в городе Алматы. Тем не менее, печать квитанций – это не основное направление компании, в основном она предлагает услуги разработки, сопровождения и организации информационных систем, а также IT-консалтинга и IT-аудита. Более половины сотрудников компании – IT специалисты. Единый расчетный центр в городе Алматы функционирует с 1998 года и позволяет горожанам получать консолидированную информацию в самом удобном для них виде:

- на бумажном носителе;
- по электронной почте;
- в Личном кабинете потребителя через интернет.<sup>1</sup>

### 1.2. Термины и аббревиатуры

Л.с. – лицевой счет;

Группа л.с. – лицевые счета, объединенные в группу (обычно в группу объединяются лицевые счета, входящие в единую систему теплоснабжения, если иное, то это указывается в контексте);

ПУ – прибор учета;

ИПУ – индивидуальный прибор учета;

ОПУ – общий прибор учета;

ЭУ – элеваторный узел;

ОПУ ЭУ – общий прибор учета, привязанный к элеваторному узлу, т.е. общедомовой прибор учета;

Основное потребление ОПУ ЭУ – потребление по ОПУ ЭУ, введенные на основе показаний ОПУ;

Сальдо – разница между начислением и оплатой;

Кредит – расход средств;

Дебет – приход средств;

### 1.3. Чем занимается компания

Непосредственными клиентами компании являются поставщики коммунальных услуг, такие как АлматыЭнергосбыт, Алматинские Тепловые



Сети, АлматыГаз, Астана ЕРЦ, и другие. Алматы – это один из немногих городов Республики Казахстан, в котором потребитель получает только одну квитанцию за коммунальные услуги. В других городах каждый владелец квартиры получает несколько квитанций – одна от поставщика электричества, вторая – за телефон, третья – за газ, четвертая за воду и так далее. Соответственно, каждый поставщик коммунальных услуг имеет свою базу данных, не связанную с остальными поставщиками. В случае изменения количества проживающих, потребитель коммунальных услуг должен в каждой коммунальной компании написать заявление об изменении, что, конечно же, очень неудобно.

Благодаря деятельности АЛСЕКО в Алматы консолидированная база данных, которая упрощает и удешевляет для потребителей оплату коммунальных услуг. Например, относительно недавно горожане получили возможность отправить данные о состоянии счетчиков – электрических, горячей и холодной воды. Раньше эту информацию поставщики коммунальных услуг получить могли лишь имея немалый штат контролеров, которые должны были обойти каждый дом и квартиру. Теперь же штат контролеров значительно сократился, что позволило поставщикам коммунальных услуг оптимизировать свои затраты.

Чтобы получить дополнительные возможности в программном обеспечении, поставщики готовят Техническое Задание на доработку программного обеспечения. Чтобы Техническое Задание соответствовало истинным пожеланиям заказчика, нужно помочь найти общий язык между рядовыми пользователями и IT специалистами, поэтому в АЛСЕКО есть целый отдел управления информационными системами. Работающие в нем менеджеры направления уточняют требования Технического Задания у заказчиков, и согласовывают его с программистами. Когда Техническое Задание согласовано, программист приступает к его реализации. Но после написания кода его нельзя просто сразу установить – для обеспечения качества программного обеспечения в АЛСЕКО есть целый отдел тестирования качества. Тестировщик проверяет доработку на соответствие Техническому заданию, проверяет программное обеспечение на наличие всех возможных ошибок и несоответствий. Тестировщик будет возвращать доработку программисту назад до тех пор, пока все найденные им ошибки не будут исправлены. Если же ошибок в программном обеспечении нет, доработка отправляется администраторам баз данных, поскольку почти у всех сотрудников АЛСЕКО нет доступа к производственной базе данных – доступ строго разграничен.

Таким образом, основной работой АЛСЕКО является изменение программного комплекса, которым пользуются поставщики коммунальных услуг. Помимо этого АЛСЕКО занимается выполнением оперативной работой обработки данных:

- Обработкой информации по платежам населения поставщикам коммунальных услуг;

- Расчёт сумм начисления по оплате за услуги по установленным тарифам;
- Печать счетов-квитанций потребителям услуг;
- Формирование ведомостей распределения платежей по банкам;
- Ведение сальдо по оплате за услуги;
- Предоставление информации для поставщиков услуг по поступившим платежам:
- Получение от поставщиков услуг информации по начислениям и корректировкам:
- Корректировка данных.

Производственный отдел АЛСЕКО обеспечивает заключение типовых договоров с КСК, ведение архива, работу с населением по оплаченным квитанциям, подготовку дополнительной информации по письмам поставщиков коммунальных услуг, предоставление необходимых отчетов заказчикам и получение от них информации для корректировок в базе данных.<sup>2</sup>

#### 1.4. Анализ предметной области

В компании АЛСЕКО очень много проектов, для которых ведутся свои разработки. Такие как газ, теплоэнергия, комплекс, электроэнергия, печать и т.п. На данный момент в основном все то, что связано с электроэнергией. Но в этом году компания приняла новый курс и решила стать обособленной IT-компанией.

Мой проект пока неотрывно связан с теплоэнергией, так как именно для это направление очень обширное, имеет много тонкостей, особенностей, которые необходимо учитывать. Для «Теплоэнергии» имеются свои справочники потребителей, как физических, так и юридических лиц, справочник об индивидуальных и общих приборах учета (ИПУ, ОПУ), об элеваторных узлах, присоединенных к дому, об операциях, которые поставщики совершали и т.п.

Самое частое, что обычно запрашивают заказчики – это отчеты и выборки по каким-то определенным параметрам. В программе ISASHeat имеются оперативные и итоговые отчеты, реализованы они через процедуры. Отчеты параметризованы: по выбранным бригадам, категориям, меткам и т.д. Используется хранилище данных – итоговые данные подготавливаются и сохраняются заранее – просмотр и выдача происходят почти мгновенно, никаких часовых, получасовых, пятиминутных ожиданий. Так же есть возможность конвертации отчетов в Excel формат для удобства расчета, просмотра данных. Группировка, фильтрации, детализация так же присутствуют.

## 1.5. Техническое задание

### 1.5.1. Общие сведения, назначение и цели разработки

Цель работы - написать процедуры и создать интерфейс для запуска тестирования.

Основной задачей является написание программы, которая по указанным параметрам будет запускать процедуру тестирования и отправлять результаты на почту и/или выводить на экран.

Требования:

- Программа "Autotests" должна иметь поля для выбора сервера, поле ввода e-mail, возможность выбора даты, указание проверки по физическим и юридическим лицам, список доступных блоков тестирования и запуск выполнения тестов;
- Список серверов должен загружаться из внешнего файла. Осуществлять проверки существует ли сервер, если ли доступ на подключение к нему, существует ли нужная процедура;
- Поле для ввода e-mail является не обязательным, если выбран вывод результат на экран;
- При выборе дат по умолчанию указан текущий период, так же есть возможность выбрать начало другого периода;
- Предусмотреть возможность запуска тестирования по физическим и/или юридическим лицам. Если ни один из вариантов не выбран, выводить ошибку при запуске тестирования;
- Должна быть возможность выбора блоков тестирования;
- При выборе сервера автоматически должна загружаться таблица с существующими блоками тестирования на нем;
- После запуска тестирования на экране должна появиться панель с надписью: "Пожалуйста, ждите" и время начала тестирования;
- При выводе результатов тестирования подсвечивать цветом тесты, которые пришли с ошибкой. Если задана почта для отправки результатов, то результаты тестирования должны отправляться на указанную почту;
- При выводе результатов на экран после каждого запуска тестирования результирующая таблица должна обновлять данные;
- Программа должна быть понятной, удобной и простой.

Пользователь вводит данные, выбирает необходимые параметры, которые передаются в процедуру в качестве аргументов.



## 1.5.2 Характеристика отчетов и справочников

Для выполнения поставленных задач необходимо проанализировать следующие объекты:

Отчет «Итоговый по бригадам» – вывод информации по юридическим лицам в разрезе договоров, в разрезе объектов, в разрезе договоров по тарифам.

Отчет «Итоговый отчет по физ. лицам» – вывод информации по лицевым счетам.

Отчет «Отчет по срокам задолженности» – вывод задолженности физических и/или юридических лиц за заданный период.

Отчет «Итоговый отчет по договорам» - вывод детальной информации по договорам.

Отчет «Отчет для загрузки в 1С Бухгалтерию» - отчет по начислениям, перерасчетам и корректировкам, сводный по дебиторской задолженности, сводный по начислению дохода.

База данных ISASHeat со всеми входящими в нее таблицами, функциями и процедурами.

Ввод данных осуществляется на двух языках –русский и английский.

## 1.5.3 Состав и сроки работ по созданию системы

- а) Разработка объектов по п. 2 настоящего ТЗ – 5 рабочих дней;
- б) Конвертация начальных данных – 20 рабочих дней;
- в) Написание необходимых процедур – 10 рабочих дней;
- г) Создание интерфейса программы – 15 рабочих дней
- д) Тестирование приложения - 4 рабочих дня;
- е) Внедрение в систему – 1 рабочий день.

## 1.5.4 Порядок контроля и приемки системы

- а) Тестовая эксплуатация системы;
- б) Нахождения ошибок при разработке и внедрении;
- в) Исправление ошибок, если таковые имеются;
- г) Внедрение системы в эксплуатацию;
- д) Техническое сопровождение;

### 1.5.5 Требования к безопасности

а) Не разглашение данных о физических и юридических лицах.

б) Заказчик не имеет права, распространять данную систему вне предприятия.

в) Выполнение всех законодательно установленных требований к хранению персональных данных.

### 1.6. Технологии, используемые для создания программного обеспечения для автоматизации тестирования

В качестве системы управления базами данных используется Microsoft SQL Server - одна из наиболее используемых реляционных систем управления базами данных.

Ее преимущества:

- Microsoft SQL Server значительно выделяется на фоне своего основного конкурента – Oracle своим ценообразованием, а на фоне более дешевых и open source – своим быстродействием и функциональностью.

- SQL Server позволяет строить кластерные отказоустойчивые решения, которые работают в режиме 24x7 при минимальных рисках потери информации.

- Безусловно, благодаря огромному количеству технической информации, платным и бесплатным курсам и большой распространенностью Microsoft SQL Server является самым простым в поддержке и сопровождении.

- Microsoft SQL Server имеет огромный задел в масштабируемости, позволяя удовлетворить потребности от маленьких компаний до транснациональных концернов.

- Microsoft SQL Server позволяет распределять ресурсы между критичными для бизнеса процессами и рутинными ресурсоемкими заданиями, что позволяет сократить затраты на серверное оборудование.

- Реализованный в Microsoft SQL Server подход является классическим доступом на ролевом подходе и доступом на уровне объектов. Эта комбинация является необходимой для построения надежной и безопасной системы управления данными.

Sql Server Management Studio - программа из пакета Microsoft SQL Server 2005 и более новых версий для изменения, конфигурирования, управления и администрирования над всеми компонентами Microsoft SQL Server. Утилита имеет редактор для скриптов и графический интерфейс, с помощью которого можно управлять всеми компонентами сервера – базами данных, таблицами, хранимыми процедурами, пользователями и x группами, триггерами, и

прочими компонентами. Является основным инструментом разработчиков и администраторов баз данных.

Sql Assistants – инструментарий, необходимый разработчикам и, естественно, администраторам баз данных. В основном используется для работы над проектами любой сложности, включая работу с малыми базами данных до огромных корпоративных систем.

Microsoft Visual Studio Professional 2015 - это программная среда, используемая в разработке программ, приложений для операционной системы Windows. Эти программы могут являться как консольными, так имеющим свой графический интерфейс.

Microsoft Visual Studio имеет интуитивный стиль кодирования. По умолчанию среда разработки будет форматировать код при его вводе, каждый раз автоматически добавляя нужные отступы, слова и применяя выделения элементов разными цветами, например, различных комментариев, либо классов, объектов. Эти самые незначительные дополнения делают код намного удобным и пригодным для чтения и, естественно, в таком коде будет легко найти ошибку. Все параметры форматирования так же можно самим настраивать по удобности использования, это очень помогает в ситуациях, когда программист привык к другому расположению скобок в своем коде.

Чем же еще отличается Visual Studio? Он имеет более высокую скорость разработки. Почти все функциональные возможности этой среды разработки направлены на то, чтобы помогать разработчику делать свою работу как можно быстрее. Для этого имеются удобные функции, такие как IntelliSense (он может находить ошибки и предлагать наиболее подходящие правильные варианты), функции поиска и замены (которая позволяет отыскивать ключевые слова. Может найти искомое слово в одном документе/файле, либо же начать поиск по всему проекту) и функции автоматического добавления и удаления комментариев (по бокам появляются знаки «плюса» и «минуса», которые могут скрывать блоки кода под этими комментариями), позволяют разработчику создавать программы довольно быстро и качественно.

Возможности отладки. Visual Studio предлагает свои инструменты для отладки, которые являются самых удобным и наиболее используемым средством для нахождения загадочных ошибок и диагностирования довольно странного поведения в создаваемом проекте. Программист может выполнять свой код построчно раз за раз, устанавливать точки останова, в любое время просматривать необходимую ему информацию, которая будет записана в память. Таким образом можно легко проверять, как выполняются методы, какие переменные в них попадают, какие являются выходными данными, где именно происходит та или иная ошибка.

Для графического интерфейса использовался Windows Forms.

Windows Forms – интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя. В основном используется для написания оконного приложения. Удобен в использовании и понимании. Абсолютно не сложен в реализации приложений.



Преимущества Windows Forms представлены на рисунке 1.

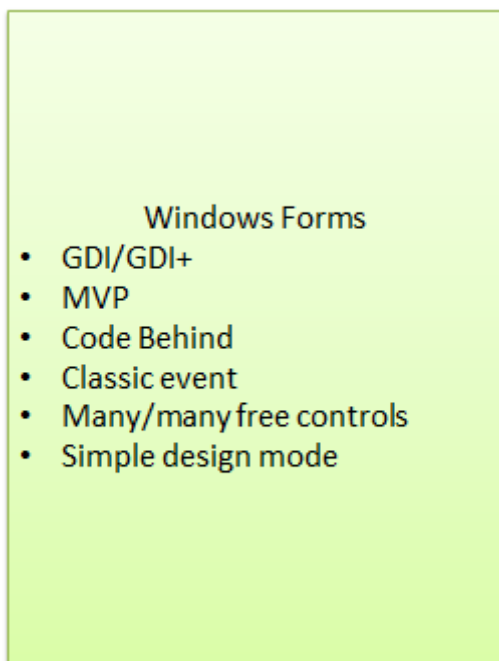


Рисунок 1. Преимущества Windows Form

## 2 Практическая часть

### 2.1. UML-диаграммы

Создание модели приложения, которое необходимо разработать, всегда должна начинаться до создания либо обновления самого приложения. Это нужно для того, чтобы яснее представить, что же из себя представляет задача, а также для взаимодействия команды разработчиков, занимающихся данным проектом и для взаимопонимания с заказчиком. Естественно это помогает понять, как архитектурно должен быть реализован проект до начала его кодирования. Для это прекрасно подходит унифицированный язык моделирования (UML), используемый для визуализации, классификации, спецификации и документирования программный продуктов.

Первая диаграмма, разработанная для программы, это диаграмма деятельности (см. рисунок 1)

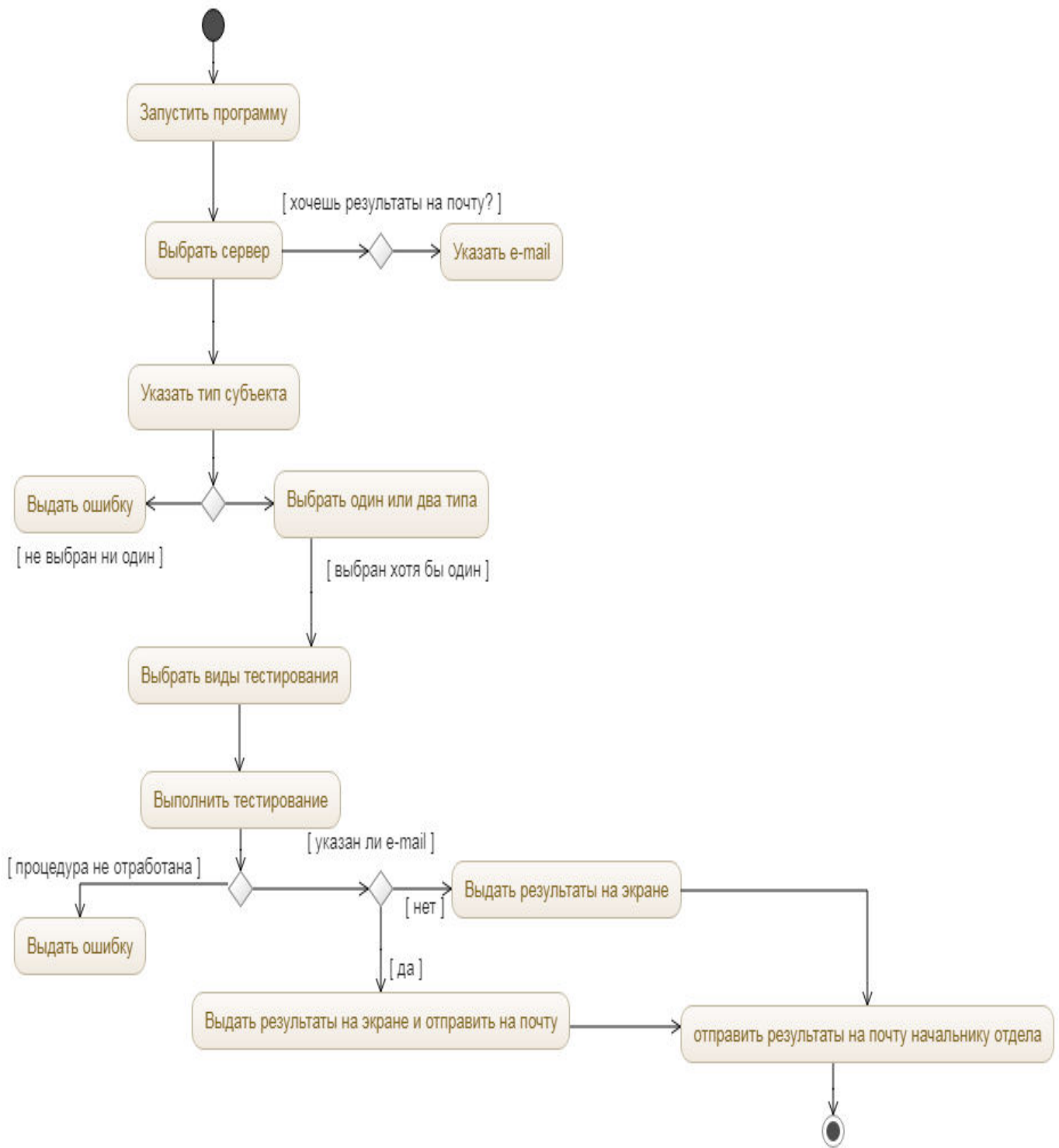


Рисунок 1. Диаграмма деятельности

Вторая диаграмма - диаграмма вариантов использования (см. рисунок 2 )

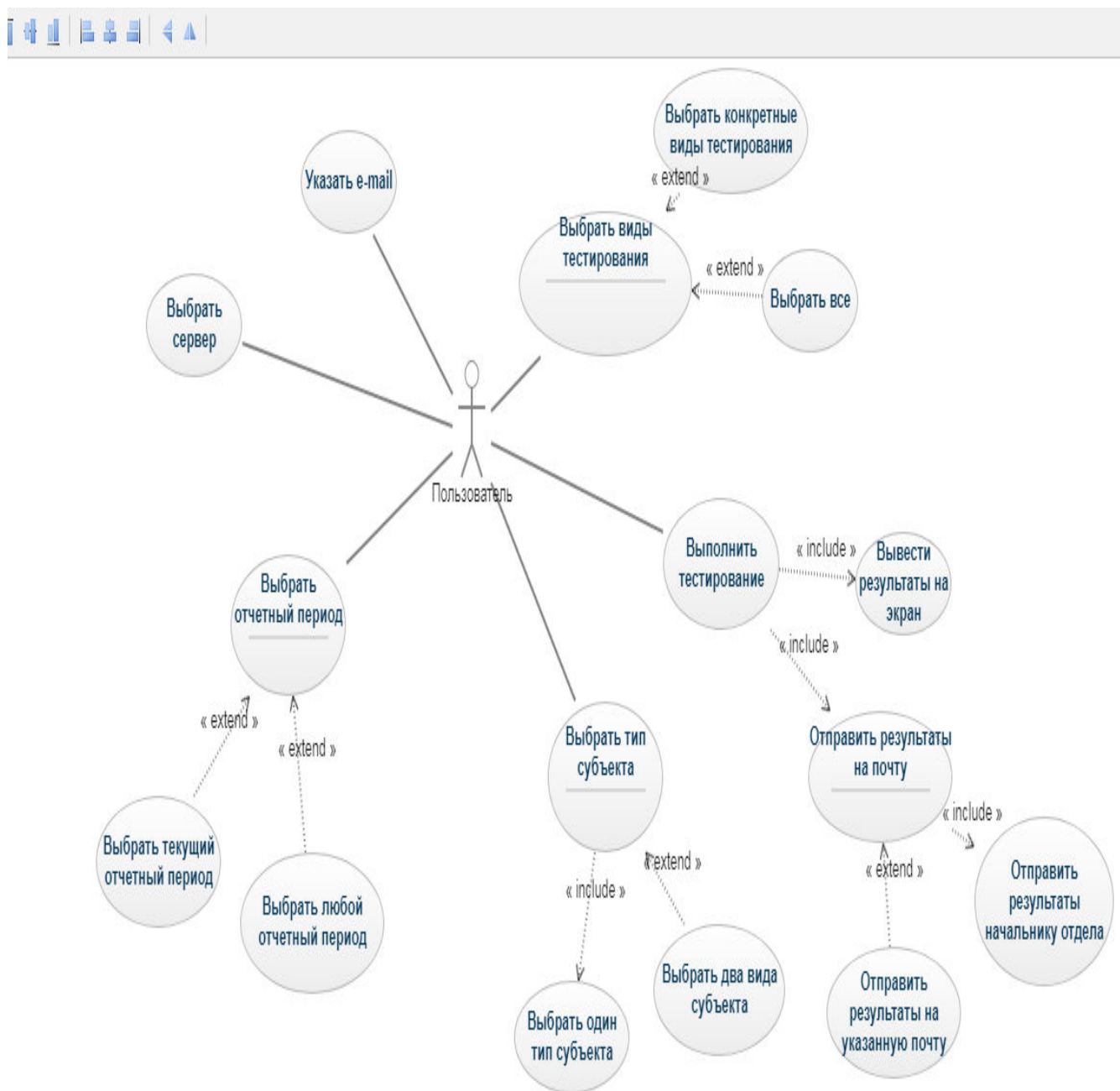


Рисунок 2. Диаграмма вариантов использования

Следующие диаграммы – это логическая и физическая модели базы данных.

Логический уровень отображает абстрактный взгляд на информацию, в нем все данные представляются так, как они должны выглядеть в реальном мире, как мы их можем представлять, понимать и называются они на нам

понятном языке, например, «Человек», «Сотрудник», «Работа». Объекты модели, находящиеся на логическом уровне, называются сущностями. А какие-либо особенности этого объекта, которые его описывают, – атрибуты. Логическая модель данных является общим описанием какой-либо предметной области и поэтому может быть создана на основе любой другой логической модели. Эта модель из-за своих особенностей является универсальной и не имеет никаких связей с конкретной реализацией СУБД.

Физическая модель данных отличается тем, что, наоборот, имеет зависимость от конкретной реализации СУБД. В физической модели содержится информация о всех сущностях базы данных и их атрибутах. Информация о типе данных хранимой информации в таблице, о количестве символов для хранения данных, является ли какой-либо атрибут первичным или внешним ключом, может ли он быть пустым (ISNULL). Одной и той же логической

Одну логическую модель могут реализовывать несколько разных физических моделей. В физической модели мы должны описывать всю имеющуюся информацию о конкретных физических объектах - таблицах, процедурах, колонках, функций, индексах и т. д.

Логическая модель должна быть понятна для всех, начиная от заказчика и заканчивая самим программистом. Физическая модель же углубляется в детальность и поэтому может быть понятна только программисту.

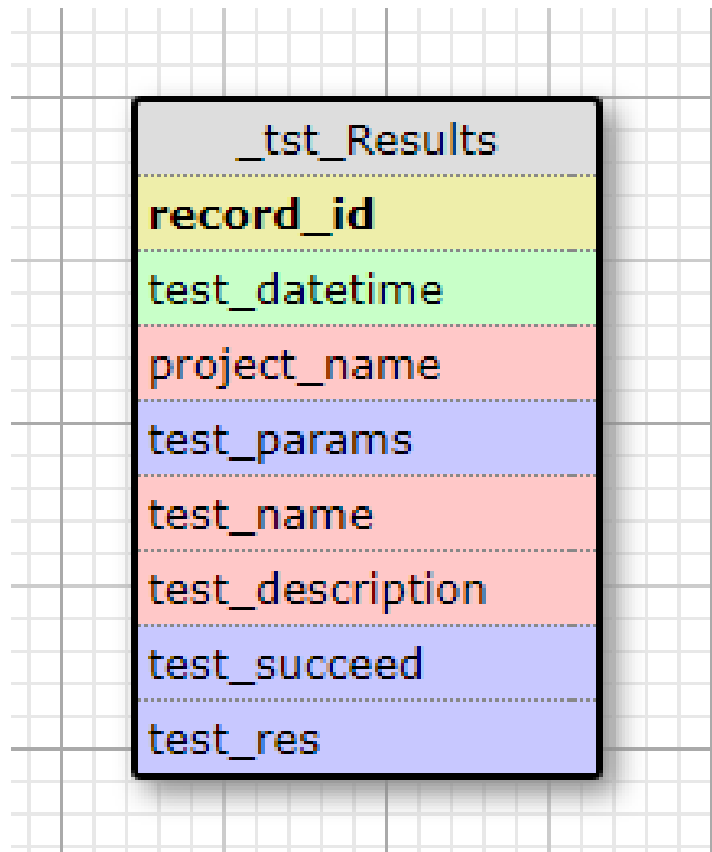


Рисунок 3. Логическая модель базы данных

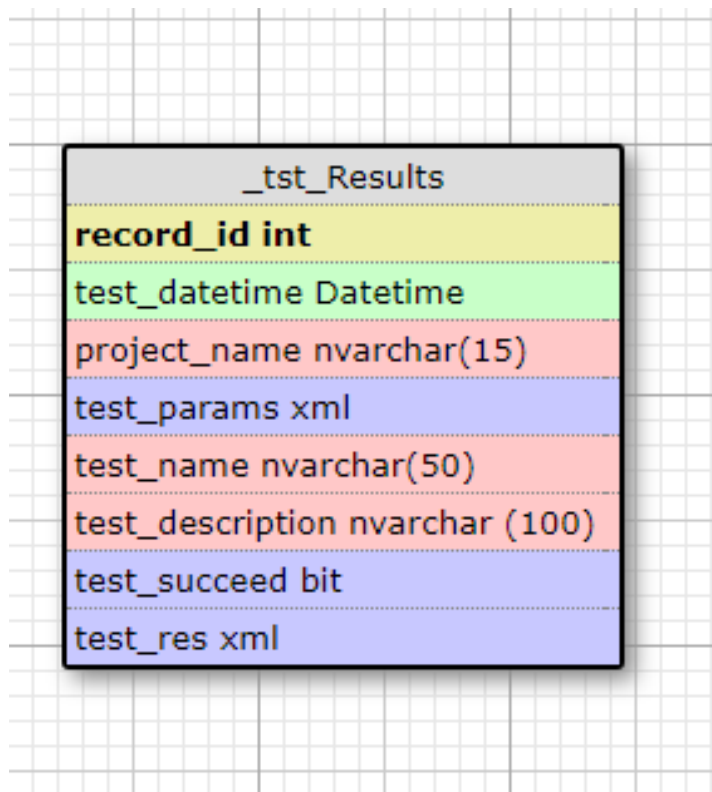


Рисунок 4. Физическая модель базы данных

### 3 Проектная часть

#### 3.1. Интерфейс приложения

Для дальнейшей работы в программе Visual Studio – Windows Forms был разработан макет создаваемого приложения. Были выставлены все необходимые компоненты, все таблицы, кнопки. Интерфейс был согласован с менеджером проекта (см. Рисунок 5).

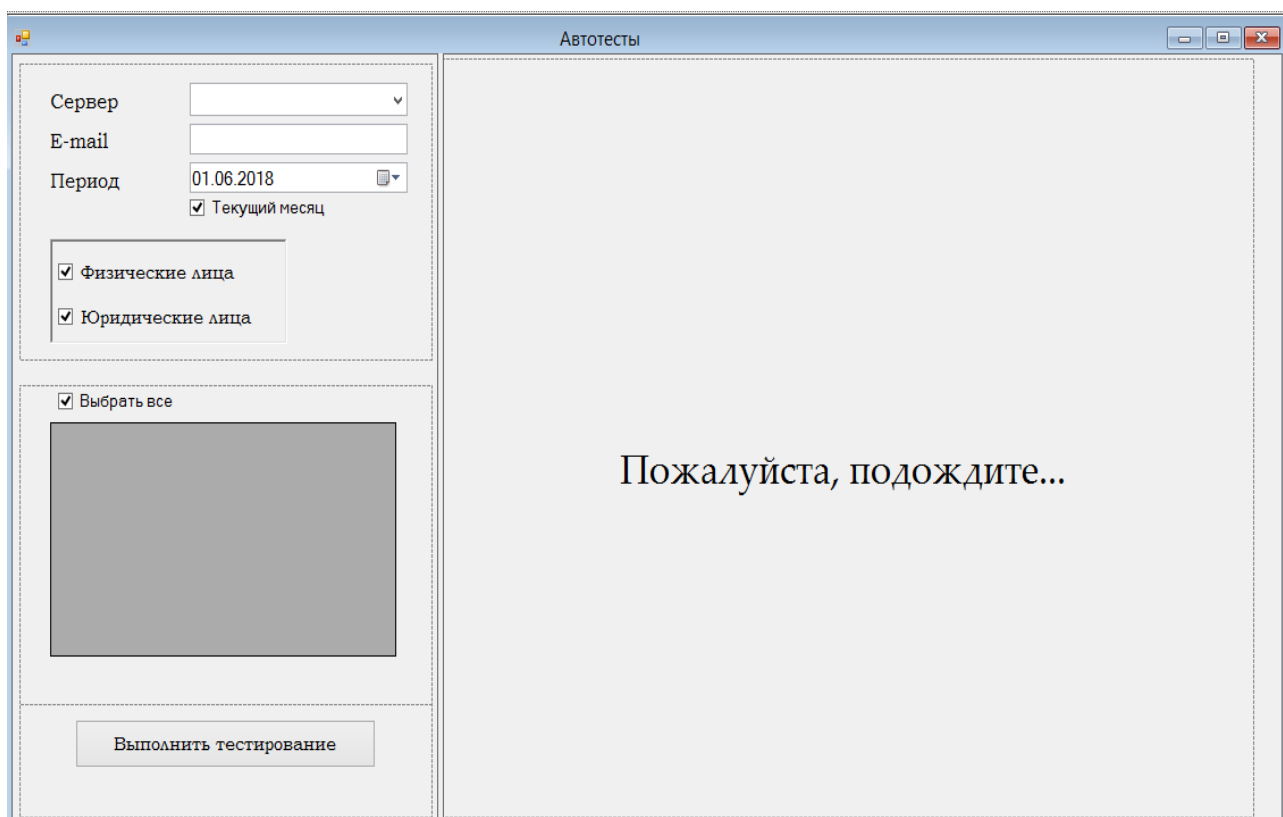


Рисунок 5. Макет приложения

Реализован выбор тестовых серверов для работы как программиста, так и тестировщика. Можно указать тип субъекта для проверки, физические или же юридические лица, так же надо указать блоки для тестирования. По умолчанию и типы субъекта и блоки тестирования уже все выбраны, по желанию можно изменить параметры (см Рисунок 6).



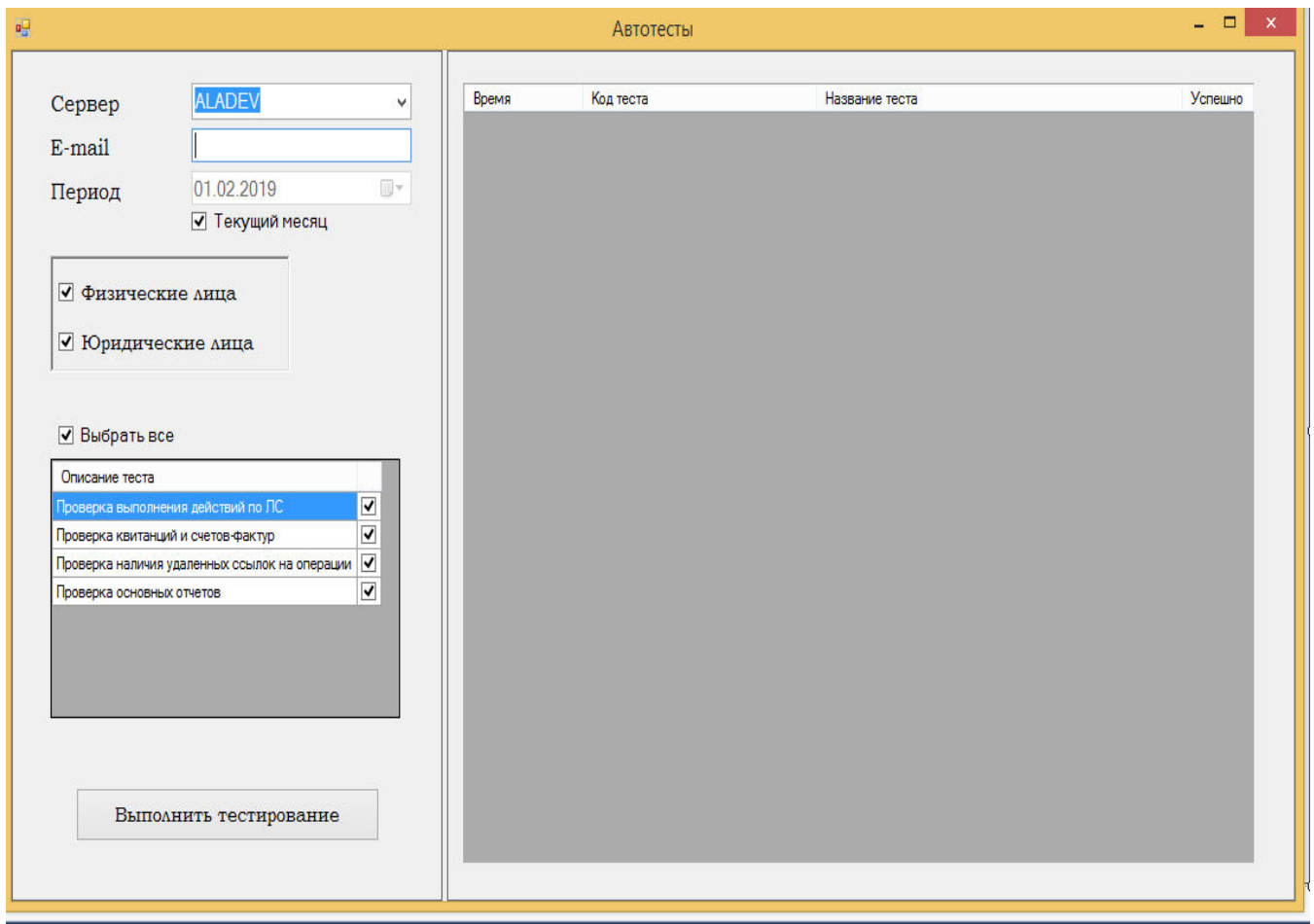


Рисунок 6. Выбор необходимых параметров

Для начала автотестирования можно изменить отчетный период. Отчетный период начинается с первого числа каждого месяца, поэтому при нажатии на любое число месяца, по дефолту выставится начало месяца. Можно оставить текущий отчетный период (см. Рисунок 7).

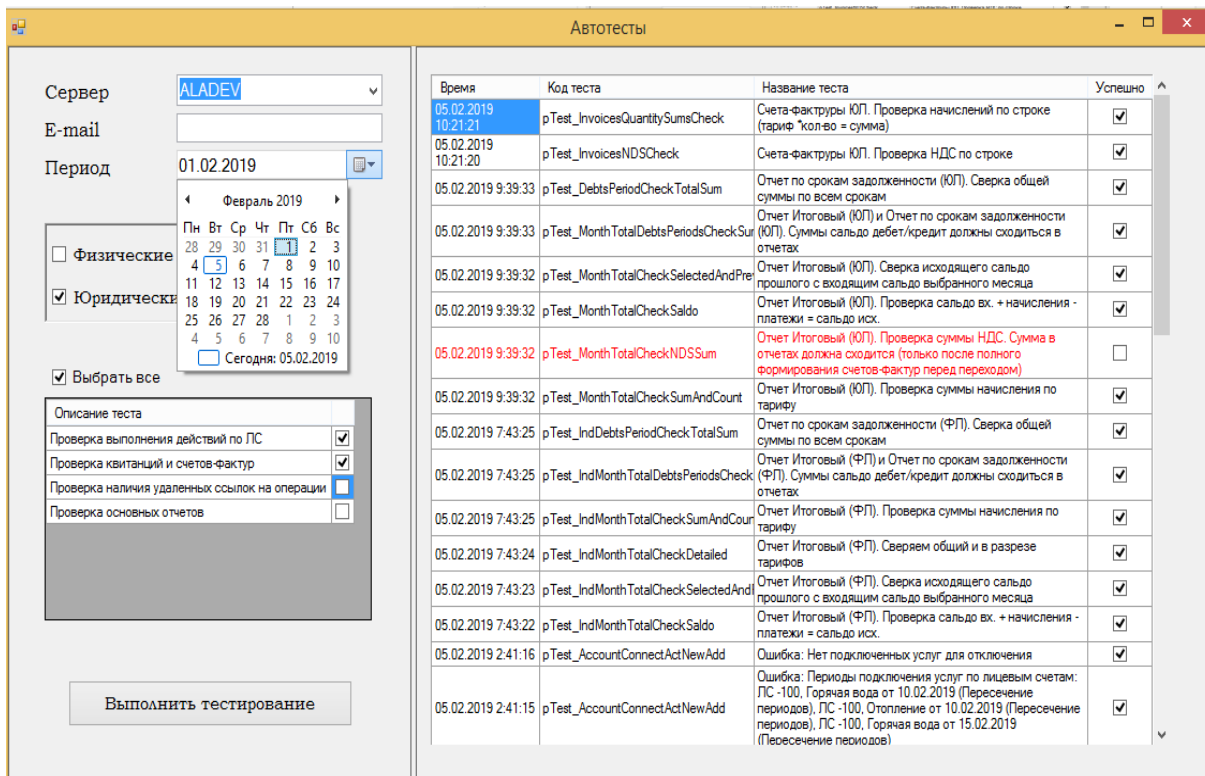


Рисунок 7. Выбор необходимой даты

Перед началом тестирования появляется MessageBox с предупреждением «Желаете ли продолжить тестирование?» (см. Рисунок 8). Он нужен для того, чтобы пользователь проверил, все ли данные указал верно. В основном тестирование может занимать от двух до шести часов (если тестировать основные отчеты, еще и включая физические лица).

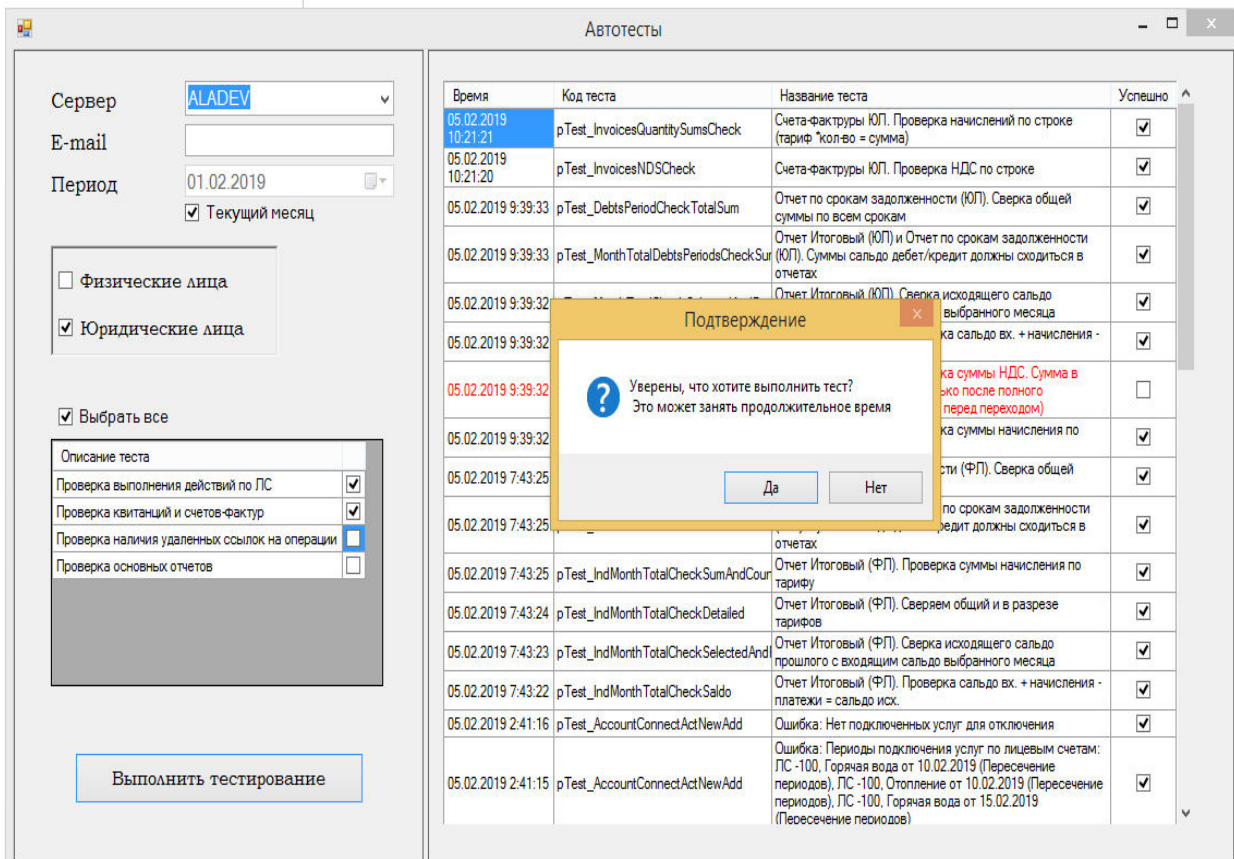


Рисунок 8. Проверка

Результаты выводятся на экран (см. Рисунок 9). Красным выделены тесты, которые закончились неудачно и в них присутствует расхождение с основным отчетом. На них внимание нужно обращать в первую очередь и начать незамедлительно их править.

Время	Код теста	Название теста	Успешно
05.02.2019 10:21:21	pTest_InvoicesQuantitySumCheck	Счета-фактуры КУП. Проверка начислений по строке (параметр 'Колво' = сумма)	✓
05.02.2019 10:21:20	pTest_InvoicesNDSCheck	Счета-фактуры КУП. Проверка НДС по строке	✓
05.02.2019 9:39:33	pTest_DebitsPeriodCheckTotalSum	Отчет по срокам задолженности (КУП). Сверка общей суммы по всем срокам	✓
05.02.2019 9:39:33	pTest_MonthTotalDebitsPeriodCheckSum	Отчет Итоговый (КУП) и Отчет по срокам задолженности (КУП). Суммы сальдо дебет/кредит должны сходиться в отчетах	✓
05.02.2019 9:39:32	pTest_MonthTotalCheckSelectedAndPer	Отчет Итоговый (КУП). Сверка исходящего сальдо прошлого с входящим сальдо выбранного месяца	✓
05.02.2019 9:39:32	pTest_MonthTotalCheckSaldo	Отчет Итоговый (КУП). Проверка сальдо вк. + начисления - платежи = сальдо иск.	✓
05.02.2019 9:39:32	pTest_MonthTotalCheckNDSSum	Отчет Итоговый (КУП). Проверка суммы НДС. Сумма в отчетах должна сходиться (только после полного формирования счетов-фактур перед периодом)	✗
05.02.2019 9:39:32	pTest_MonthTotalCheckSumAndCount	Отчет Итоговый (КУП). Проверка суммы начисления по тарифу	✓
05.02.2019 7:43:25	pTest_IndDebitsPeriodCheckTotalSum	Отчет по срокам задолженности (ФП). Сверка общей суммы по всем срокам	✓
05.02.2019 7:43:25	pTest_IndMonthTotalDebitsPeriodCheck	Отчет Итоговый (ФП) и Отчет по срокам задолженности (ФП). Суммы сальдо дебет/кредит должны сходиться в отчетах	✓
05.02.2019 7:43:25	pTest_IndMonthTotalCheckSumAndCount	Отчет Итоговый (ФП). Проверка суммы начисления по тарифу	✓
05.02.2019 7:43:24	pTest_IndMonthTotalCheckDetailed	Отчет Итоговый (ФП). Сверка общей и в разрезе тарифов	✓
05.02.2019 7:43:23	pTest_IndMonthTotalCheckSelectedAndPer	Отчет Итоговый (ФП). Сверка исходящего сальдо прошлого с входящим сальдо выбранного месяца	✓
05.02.2019 7:43:22	pTest_IndMonthTotalCheckSaldo	Отчет Итоговый (ФП). Проверка сальдо вк. + начисления - платежи = сальдо иск.	✓
05.02.2019 2:41:16	pTest_AccountConnectActNewAdd	Ошибка: Нет подключенных услуг для отключения	✓
05.02.2019 2:41:15	pTest_AccountConnectActNewAdd	Ошибка: Периоды подключения услуг по заданным счетам: ПС -100. Горячая вода от 10.02.2019 (Периодичность периода); ПС -100. Отопление от 10.02.2019 (Периодичность периода); ПС -100. Горячая вода от 15.02.2019 (Периодичность периода)	✓

Рисунок 9. Результаты тестирования

Если указать свой e-mail, то результат тестирования придет и туда, помимо выведения на экран. Первыми строками при наличии ошибок будут именно те, которые о них и рассказывают. После них идет блок тестов, выполненных успешно (см. рисунок 10).

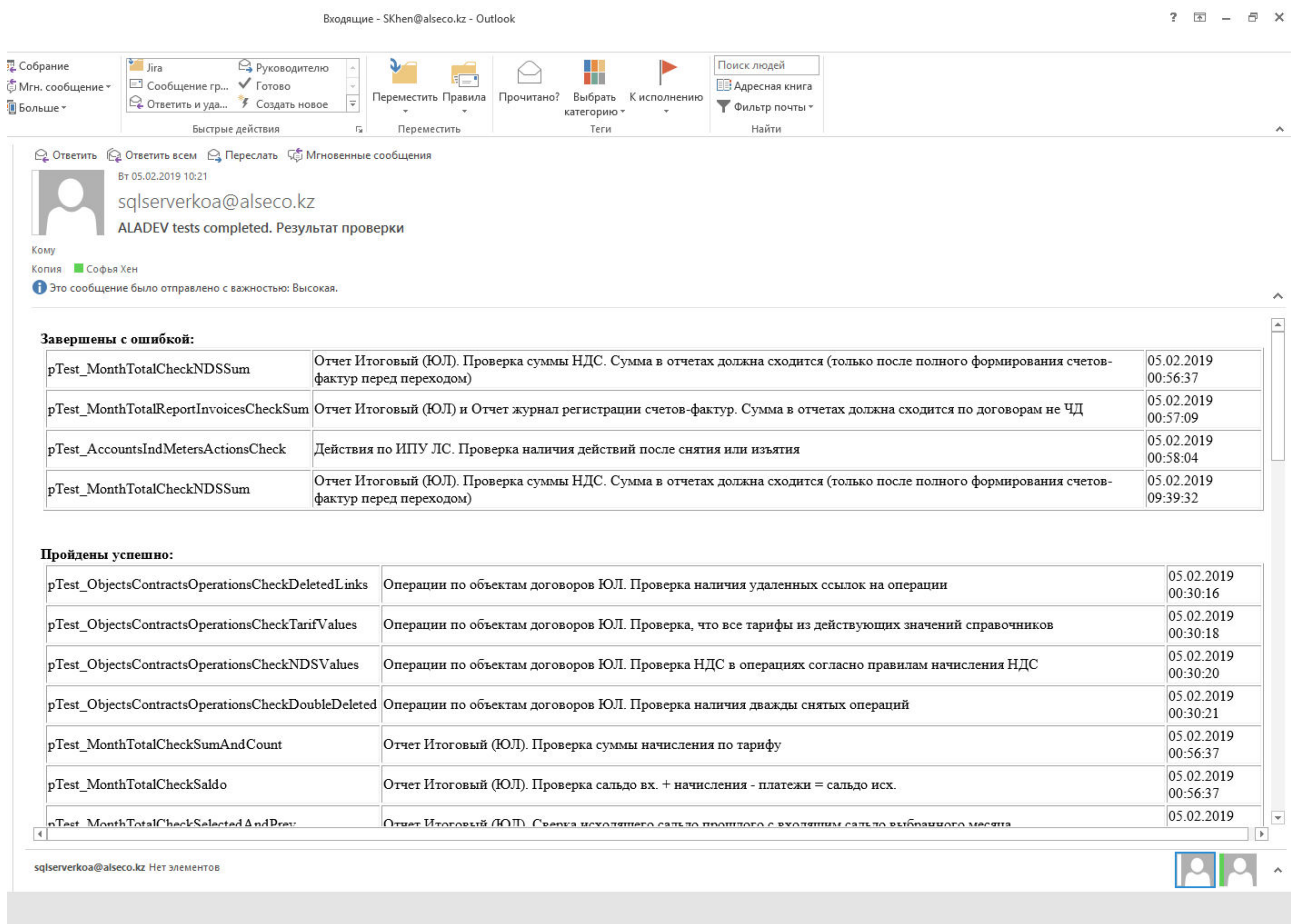


Рисунок 10. Результаты, пришедшие на почту

Если не выбран какой-то из нужных параметров, то происходит проверка и при наличии несоответствий появляется предупреждение (см. Рисунок 11) и в фокусе будет тот блок, где именно не заполнены данные. В этом блоке сразу появляется предупреждающий значок.

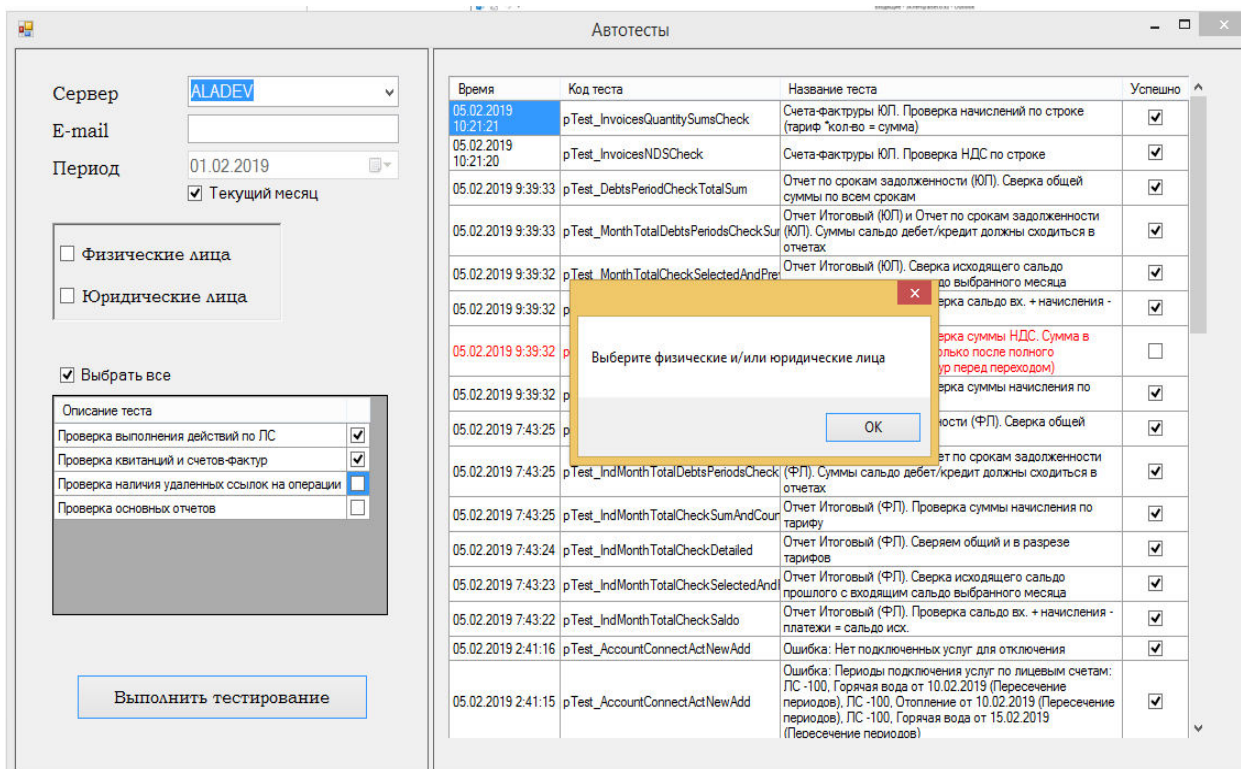


Рисунок 11. Предупреждение



## ЗАКЛЮЧЕНИЕ

В ходе выполнения данной работы разработана и реализована программная система, позволяющая следить за состоянием целостности и непротиворечивости данных в созданных и создаваемых отчетах, а также сохранять информацию в базе данных с возможностью поиска и просмотра итога автотестирования, отправлять на почту результаты тестирования.

Применение системы направлено на повышение эффективности работы отдела тестирования программ и продуктов, а также на уменьшение ошибок программистов при разработке новых отчетов.

При разработке были проведены следующие работы:

1. Ознакомление с предметной областью;
2. Анализ деятельности компании АЛСЕКО;
3. Изучение реляционной базы данных Microsoft SQL Server;
4. Изучение системы Delphi XE 10, с помощью которых были созданы итоговые отчеты;
5. Проектирование и реализация автоматизированной системы “Автотестирование”;
6. Опытная и грамотная эксплуатация разработанной системы.

В данный момент система внедрена, продолжается ее сопровождение и доработка. В процессе эксплуатации получены положительные отзывы от специалистов отделов разработки и поддержки программного обеспечения (ОРПО) и тестирования качества (ОТК) АЛСЕКО. Для дальнейшего сопровождения проекта планируется добавление в систему возможностей вывода на экран графиков стабильности работы автотестов.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Линн Бейли, редакция под руководством Матвеева Е. Изучаем SQL. – Санкт-Петербург: Питер, 2012г. –С.77-471.
2. Э.Стиллмен, Дж. Грин Программирование на С# и на платформе .NET 4.0, 1 издание. – Санкт-Петербург: Питер, 2011. –С.150-300.
3. Э.Стиллмен, Дж. Грин Программирование на С# и на платформе .NET 4.0, 2 издание. – Санкт-Петербург: Питер, 2011. –С.20-63.
4. Э.Фримен, Э.Фримен, К.Сьерра, Б.Бейтс Паттерны проектирования. – Санкт-Петербург: Питер, 2011г. – С.15-124.
5. Д.Пилон, Р.Майлз Управление разработкой ПО. – Санкт-Петербург: Питер, 2011. –С.15-101.
6. Э.Стиллмен, Дж.Грин Изучаем С#, 1-ое издание. – Санкт-Петербург: Питер, 2011. –С.400-659.
7. Э.Стиллмен, Дж.Грин Изучаем С#, 2-ое издание. – Санкт-Петербург: Питер, 2011. –С.9-125.
8. Р. Мартин Чистый код: создание, анализ и рефакторинг. – Санкт-Петербург: Питер, 2011. –С.5-134.
9. Сайт о программировании Metanit.com, Глава 2. Основы программирования на С# [электронный ресурс]. – URL: <https://metanit.com/sharp/tutorial/2.1.php>
10. Сайт о программировании Metanit.com, Глава 3. Классы. Объектно-ориентированное программирование [электронный ресурс]. – URL: <https://metanit.com/sharp/tutorial/3.1.php>
11. Сайт о программировании Metanit.com, Глава 4. Обработка исключений [электронный ресурс]. – URL: <https://metanit.com/sharp/tutorial/2.14.php>
12. Сайт о программировании Metanit.com, Глава 5. Интерфейсы [электронный ресурс]. – URL: <https://metanit.com/sharp/tutorial/3.9.php>
13. Сайт о программировании Metanit.com, Глава 6. Делегаты, события и лямбды [электронный ресурс]. – URL: <https://metanit.com/sharp/tutorial/3.13.php>
14. Сайт о программировании Metanit.com, Глава 7. Дополнительные возможности ООП в С# [электронный ресурс]. – URL: <https://metanit.com/sharp/tutorial/3.25.php>
15. Сайт о программировании Metanit.com, Глава 9. Коллекции [электронный ресурс]. – URL: <https://metanit.com/sharp/tutorial/4.1.php>
16. Сайт о программировании Metanit.com, Глава 10. Работа с потоками и файловой системой [электронный ресурс]. – URL: <https://metanit.com/sharp/tutorial/5.1.php>
17. Сайт о программировании Metanit.com, Глава 12. Работа со строками [электронный ресурс]. – URL: <https://metanit.com/sharp/tutorial/7.1.php>

18. Сайт о программировании Metanit.com, Глава 12. Работа со строками  
Создание графического приложения [электронный ресурс]. – URL:  
<https://metanit.com/sharp/windowsforms/1.1.php>

1 См. об этом: [Электронный ресурс], URL: <https://www.alseco.kz/home>

2 См. об этом: [Электронный ресурс], URL: <https://www.alseco.kz/it-audit-consulting>

## Приложение А (обязательное)

### Текст программы

#### Главное окно

```
using System;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.IO;
using System.Threading;
using System.Windows.Forms;
namespace autotests
{
    public partial class Form1 : Form
    {
        DataTable testTypes;
        SqlDataAdapter testTypesDataAdapter;
        SqlCommand createTable;
        SqlConnection connection;
        SqlCommand command;
        SqlCommand updCom;// = new SqlCommand();
        string selectQuery;
        const string agreement = "Уверены, что хотите выполнить тест? \r\n Это
может занять продолжительное время";
        const string fileName = "serv.ini";
        public Form1()
        {
            InitializeComponent();
            {
                connection = new SqlConnection();
                updCom = new SqlCommand();
                updCom.Connection = connection;
                command = new SqlCommand();
                testTypes = new DataTable();
                createTable = new SqlCommand();
                testTypesDataAdapter = new SqlDataAdapter(selectQuery, connection);
                updCom.CommandText = "update [dbo].[#ltHelp] set [is_checked] =
@is_checked where record_id = @record_id";
                updCom.CommandType = CommandType.Text;
```

## Продолжение приложения А

```
        updCom.Parameters.Add(new SqlParameter("@record_id", SqlDbType.Int,
0, ParameterDirection.Input, 0, 0, "record_id", DataRowVersion.Original, false, null,
"", "", ""));
        updCom.Parameters.Add(new SqlParameter("@is_checked",
SqlDbType.Bit, 0, ParameterDirection.Input, 0, 0, "is_checked",
DataRowVersion.Current, false, null, "", "", ""));
    }
    GettingListOfServices();
    SetNewServer(serverList.Text);
    CreateTempTable();
}
// кнопка для начала тестирования
private void startBtn_Click(object sender, EventArgs e)
{
    DialogResult dr = MessageBox.Show(agreement, "Подтверждение",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (dr == DialogResult.Yes)
    {
        TestStart();
    }
}
// dateTimePicker указывает только на первый день месяца
private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
{
    date_begin.Value = new DateTime(date_begin.Value.Year,
date_begin.Value.Month, 1);
}
// чекбокс для выбора сразу всех блоков тестирования
private void checkBox4_CheckedChanged(object sender, EventArgs e)
{
    foreach (DataGridViewRow row in testBlocks.Rows)
    {
        DataGridViewCheckBoxCell ch =
(DataGridViewCheckBoxCell)row.Cells[3];
        ch.Value = chooseAllBlocks.Checked;
    }
}
// чекбокс для указания текущей даты
private void checkBox1_CheckedChanged(object sender, EventArgs e)
{
    date_begin.Enabled = !currentPeriod.Checked;
    if (date_begin.Enabled)
```



## Продолжение приложения А

```
date_begin.Value = new DateTime(DateTime.Now.Year,
DateTime.Now.Month, 1);
}
//Заполнение comboBox из ini файла
public void GettingListOfServices()
{
    if (File.Exists(fileName)) {
        string S1;
        StreamReader servFile = new StreamReader(fileName);
        serverList.BeginUpdate();
        while ((S1 = servFile.ReadLine()) != null)
        {
            serverList.Items.Add(S1);
        }
        servFile.Close();
        serverList.EndUpdate();
        this.serverList.SelectedIndex = 0;
    }
    else
    {
        MessageBox.Show("Файла autotest.ini не существует. ");
    }
}
//создание адаптера и код на обновление таблицы
public void DisplayBlockTypes()
{
    try
    {
        selectQuery = "select * from #ltHelp";
        testTypesDataAdapter = new SqlDataAdapter(selectQuery, connection);
        testTypesDataAdapter.Fill(testTypes);
        testBlocks.DataSource = testTypes;
        testBlocks.Columns["record_id"].Visible = false;
        testBlocks.Columns["test_name"].Visible = false;
        testBlocks.Columns["test_description"].HeaderText = "Описание теста";
        testBlocks.Columns["is_checked"].HeaderText = "";
        //команда на обновление временной таблицы
        {
            this.testTypesDataAdapter.UpdateCommand = updCom;
        }
    }
    catch (Exception ex)
```

## Продолжение приложения А

```
{
    MessageBox.Show(ex.Message);
}
}
// создание таблицы для вывода результатов на экран
public void CreateTableAllResult(SqlDataReader rrr)
{
    if (!rrr.IsClosed)
        rrr.Close();
    command.CommandType = CommandType.Text;
    command.CommandText = "select * from dbo._tst_Results order by
test_datetime desc ";
    command.Connection = connection;
    using (rrr = command.ExecuteReader())
    {
        while (rrr.Read())
        {
            DataRow row = testTypes.NewRow();
            int n = allResult.Rows.Add();
            allResult.Rows[n].Cells[0].Value = rrr["test_datetime"].ToString();
            allResult.Rows[n].Cells[1].Value = rrr["test_name"].ToString();
            allResult.Rows[n].Cells[2].Value = rrr["test_description"].ToString();
            allResult.Rows[n].Cells[3].Value = rrr["test_succeed"].ToString();
            if (!Convert.ToBoolean(allResult.Rows[n].Cells[3].Value))
            {
                this.allResult.Rows[n].DefaultCellStyle.ForeColor = Color.Red;
            }
            allResult.Rows[n].Cells[4].Value = rrr["test_res"].ToString();
        }
        rrr.Close();
    }
}
private void SetNewServer(string serverName)
{
    if (connection.State == ConnectionState.Open)
        connection.Close();
    connection.ConnectionString = "Data Source=" + serverName +
";Integrated Security=SSPI;Initial Catalog=ISASHeat";
    connection.Open();
}
//создание временной таблицы #ltHelp
public void CreateTempTable()
```

## Продолжение приложения А

```
try
{
    createTable.Connection = connection;
    createTable.CommandType = CommandType.Text;
    createTable.CommandText = "if object_id('tempdb.dbo.#ltHelp', 'u') is null
create table #ltHelp(record_id int not null, test_name nvarchar(200) not null,
test_description nvarchar(500) null, is_checked bit not null default(0), primary
key(test_name))";
    createTable.ExecuteNonQuery();
    createTable.CommandText = "exec dbo.pTest_AllStart @bHelp = 1";
    createTable.ExecuteNonQuery();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}
// запуск тестов
public void TestStart()
{
    try
    {
        if (!FizStart.Checked && !JurStart.Checked)
        {
            MessageBox.Show("Выберите физические и/или юридические лица
");
            panel1.Focus();
        }
        Validate();
        this.testBlocks.EndEdit();
        testTypesDataAdapter.Update(testTypes);
        SqlCommand startTest = new SqlCommand();
        startTest.CommandType = CommandType.StoredProcedure;
        startTest.Connection = connection;
        startTest.CommandText = "pTest_AllStart";
        startTest.Parameters.AddWithValue("@nvcRecipients", mail.Text);
        if (currentPeriod.Checked)
        {
            startTest.Parameters.AddWithValue("@dDateCalcBegin", null);
        }
        else
        {
```

## Продолжение приложения А

```
startTest.Parameters.AddWithValue("@dDateCalcBegin",
date_begin.Value);
    }
    startTest.Parameters.AddWithValue("@bFizStart", FizStart.Checked);
    startTest.Parameters.AddWithValue("@bJurStart", JurStart.Checked);
    startTest.Parameters.AddWithValue("@bHelp", false);
    SqlDataReader testResultsReader = startTest.ExecuteReader();
    CreateTeableAllResult(testResultsReader);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
// подключение при изменении сервера
private void serverList_SelectedIndexChanged(object sender, EventArgs e)
{
    try
    {
        allResult.Rows.Clear();
        allResult.Refresh();
        SetNewServer(serverList.Text);
        CreateTempTable();
        DisplayBlockTypes();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
}
```

## Продолжение приложения А

Designer.cs

```
namespace autotests
{
    partial class Form1
    {
        /// <summary>
        /// Обязательная переменная конструктора.
        /// </summary>
        private System.ComponentModel.IContainer components = null;
        /// <summary>
        /// Освободить все используемые ресурсы.
        /// </summary>
        /// <param name="disposing">истинно, если управляемый ресурс должен
        быть удален; иначе ложно.</param>
        protected override void Dispose(bool disposing)    {
            if (disposing && (components != null))    {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
        private void InitializeComponent()
        {
            System.Windows.Forms.DataGridViewCellStyle dataGridViewCellStyle3 =
new System.Windows.Forms.DataGridViewCellStyle();
            this.startBtn = new System.Windows.Forms.Button();
            this.serverList = new System.Windows.Forms.ComboBox();
            this.label1 = new System.Windows.Forms.Label();
            this.label2 = new System.Windows.Forms.Label();
            this.mail = new System.Windows.Forms.TextBox();
            this.label3 = new System.Windows.Forms.Label();
            this.date_begin = new System.Windows.Forms.DateTimePicker();
            this.currentPeriod = new System.Windows.Forms.CheckBox();
            this.FizStart = new System.Windows.Forms.CheckBox();
            this.JurStart = new System.Windows.Forms.CheckBox();
            this.chooseAllBlocks = new System.Windows.Forms.CheckBox();
            this.panel1 = new System.Windows.Forms.Panel();
            this.testBlocks = new System.Windows.Forms.DataGridView();
            this.panel2 = new System.Windows.Forms.Panel();
            this.panel4 = new System.Windows.Forms.Panel();
            this.panel3 = new System.Windows.Forms.Panel();
        }
    }
}
```

## Продолжение приложения А

```
this.allResult = new System.Windows.Forms.DataGridView();
    this.test_datetime = new
System.Windows.Forms.DataGridViewTextBoxColumn();
    this.test_name = new
System.Windows.Forms.DataGridViewTextBoxColumn();
    this.test_description = new
System.Windows.Forms.DataGridViewTextBoxColumn();
    this.test_succeed = new
System.Windows.Forms.DataGridViewCheckBoxColumn();
    this.test_res = new System.Windows.Forms.DataGridViewTextBoxColumn();
    this.panel5 = new System.Windows.Forms.Panel();
    this.panel6 = new System.Windows.Forms.Panel();
    this.panel1.SuspendLayout();
    ((System.ComponentModel.ISupportInitialize)(this.testBlocks)).BeginInit();
    this.panel2.SuspendLayout();
    this.panel4.SuspendLayout();
    this.panel3.SuspendLayout();
    ((System.ComponentModel.ISupportInitialize)(this.allResult)).BeginInit();
    this.panel6.SuspendLayout();
    this.SuspendLayout();
    //
    // startBtn
    //
    this.startBtn.Font = new System.Drawing.Font("Bookman Old Style", 11.25F,
System.Drawing.FontStyle.Italic, System.Drawing.GraphicsUnit.Point, ((byte)0));
    this.startBtn.Location = new System.Drawing.Point(76, 17);
    this.startBtn.Name = "startBtn";
    this.startBtn.Size = new System.Drawing.Size(227, 38);
    this.startBtn.TabIndex = 0;
    this.startBtn.Text = "ВЫПОЛНИТЬ ТЕСТИРОВАНИЕ";
    this.startBtn.UseVisualStyleBackColor = true;
    this.startBtn.Click += new System.EventHandler(this.startBtn_Click);
    // serverList
    //
    this.serverList.Font = new System.Drawing.Font("Microsoft Sans Serif",
11.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)204));
    this.serverList.FormattingEnabled = true;
    this.serverList.Location = new System.Drawing.Point(150, 15);
    this.serverList.Name = "serverList";
```



## Продолжение приложения А

```
this.serverList.Size = new System.Drawing.Size(182, 26);
this.serverList.TabIndex = 1;
this.serverList.SelectedIndexChanged += new
System.EventHandler(this.serverList_SelectedIndexChanged);
//
// label1
//
this.label1.AutoSize = true;
this.label1.Font = new System.Drawing.Font("Bookman Old Style", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
(byte)(204));
this.label1.Location = new System.Drawing.Point(23, 19);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(69, 20);
this.label1.TabIndex = 2;
this.label1.Text = "Сервер";
//
// label2
//
this.label2.AutoSize = true;
this.label2.Font = new System.Drawing.Font("Bookman Old Style", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
(byte)(204));
this.label2.Location = new System.Drawing.Point(23, 50);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(60, 20);
this.label2.TabIndex = 3;
this.label2.Text = "E-mail";
//
// mail
//
this.mail.Font = new System.Drawing.Font("Microsoft Sans Serif", 11.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
(byte)(204));
this.mail.Location = new System.Drawing.Point(150, 47);
this.mail.Name = "mail";
this.mail.Size = new System.Drawing.Size(182, 24);
this.mail.TabIndex = 4;
//
// label3
//
this.label3.AutoSize = true;
```

## Продолжение приложения А

```

this.label3.Font = new System.Drawing.Font("Bookman Old Style", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
    this.label3.Location = new System.Drawing.Point(23, 81);
    this.label3.Name = "label3";
    this.label3.Size = new System.Drawing.Size(70, 20);
    this.label3.TabIndex = 5;
    this.label3.Text = "Период";
    //
    // date_begin
    //
    this.date_begin.CustomFormat = "dd-MM-yyyy";
    this.date_begin.Enabled = false;
    this.date_begin.Font = new System.Drawing.Font("Microsoft Sans Serif",
11.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
    this.date_begin.Format =
System.Windows.Forms.DateTimePickerFormat.Short;
    this.date_begin.Location = new System.Drawing.Point(150, 77);
    this.date_begin.Name = "date_begin";
    this.date_begin.Size = new System.Drawing.Size(153, 24);
    this.date_begin.TabIndex = 6;
    this.date_begin.Value = new System.DateTime(2018, 6, 1, 0, 0, 0, 0);
    this.date_begin.ValueChanged +=
System.EventHandler(this.dateTimePicker1_ValueChanged);
    // currentPeriod
    this.currentPeriod.AutoSize = true;
    this.currentPeriod.Checked = true;
    this.currentPeriod.CheckState =
System.Windows.Forms.CheckState.Checked;
    this.currentPeriod.Font = new System.Drawing.Font("Microsoft Sans Serif",
9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
((byte)(204)));
    this.currentPeriod.Location = new System.Drawing.Point(150, 103);
    this.currentPeriod.Name = "currentPeriod";
    this.currentPeriod.Size = new System.Drawing.Size(126, 20);
    this.currentPeriod.TabIndex = 7;
    this.currentPeriod.Text = "Текущий месяц";
    this.currentPeriod.UseVisualStyleBackColor = true;
    this.currentPeriod.CheckedChanged +=
System.EventHandler(this.checkBox1_CheckedChanged);

```

## Продолжение приложения А

```
//  
// FizStart  
//  
this.FizStart.AutoSize = true;  
this.FizStart.Checked = true;  
this.FizStart.CheckState = System.Windows.Forms.CheckState.Checked;  
this.FizStart.Font = new System.Drawing.Font("Bookman Old Style", 11.25F,  
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,  
((byte)(204)));  
this.FizStart.Location = new System.Drawing.Point(5, 13);  
this.FizStart.Name = "FizStart";  
this.FizStart.Size = new System.Drawing.Size(163, 23);  
this.FizStart.TabIndex = 8;  
this.FizStart.Text = "Физические лица";  
this.FizStart.UseVisualStyleBackColor = true;  
//  
// JurStart  
//  
this.JurStart.AutoSize = true;  
this.JurStart.Checked = true;  
this.JurStart.CheckState = System.Windows.Forms.CheckState.Checked;  
this.JurStart.Font = new System.Drawing.Font("Bookman Old Style", 11.25F,  
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,  
((byte)(204)));  
this.JurStart.Location = new System.Drawing.Point(5, 48);  
this.JurStart.Name = "JurStart";  
this.JurStart.Size = new System.Drawing.Size(178, 23);  
this.JurStart.TabIndex = 9;  
this.JurStart.Text = "Юридические лица";  
this.JurStart.UseVisualStyleBackColor = true;  
//  
// chooseAllBlocks  
this.chooseAllBlocks.AutoSize = true;  
this.chooseAllBlocks.Checked = true;  
this.chooseAllBlocks.CheckState =  
System.Windows.Forms.CheckState.Checked;  
this.chooseAllBlocks.Font = new System.Drawing.Font("Microsoft Sans  
Serif", 9.75F, System.Drawing.FontStyle.Regular,  
System.Drawing.GraphicsUnit.Point, ((byte)(204)));  
this.chooseAllBlocks.Location = new System.Drawing.Point(6, 7);  
this.chooseAllBlocks.Name = "chooseAllBlocks";  
this.chooseAllBlocks.Size = new System.Drawing.Size(109, 20);
```

## Продолжение приложения А

```

this.chooseAllBlocks.TabIndex = 13;
this.chooseAllBlocks.Text = "Выбрать все";
this.chooseAllBlocks.UseVisualStyleBackColor = true;
this.chooseAllBlocks.CheckedChanged += new
System.EventHandler(this.checkBox4_CheckedChanged);
//
// panel1
//
this.panel1.BorderStyle = System.Windows.Forms.BorderStyle.Fixed3D;
this.panel1.Controls.Add(this.FizStart);
this.panel1.Controls.Add(this.JurStart);
this.panel1.Location = new System.Drawing.Point(27, 129);
this.panel1.Name = "panel1";
this.panel1.Size = new System.Drawing.Size(200, 82);
this.panel1.TabIndex = 14;
//
// testBlocks
//
this.testBlocks.AllowUserToAddRows = false;
this.testBlocks.AllowUserToDeleteRows = false;
this.testBlocks.AllowUserToOrderColumns = true;
this.testBlocks.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top
| System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left));
this.testBlocks.AutoSizeColumnsMode =
System.Windows.Forms.DataGridViewAutoSizeColumnsMode.AllCells;
this.testBlocks.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewAutoSizeColumnsMode.AutoSize;
this.testBlocks.Location = new System.Drawing.Point(27, 33);
this.testBlocks.Name = "testBlocks";
this.testBlocks.RowHeadersVisible = false;
this.testBlocks.Size = new System.Drawing.Size(291, 181);
this.testBlocks.TabIndex = 15;
//
// panel2
this.panel2.Controls.Add(this.label1);
this.panel2.Controls.Add(this.serverList);
this.panel2.Controls.Add(this.label2);
this.panel2.Controls.Add(this.mail);
this.panel2.Controls.Add(this.panel1);
this.panel2.Controls.Add(this.label3);

```

## Продолжение приложения А

```
this.panel2.Controls.Add(this.date_begin);
this.panel2.Controls.Add(this.currentPeriod);
this.panel2.Location = new System.Drawing.Point(12, 12);
this.panel2.Name = "panel2";
this.panel2.Size = new System.Drawing.Size(371, 243);
this.panel2.TabIndex = 18;
//
// panel4
//
this.panel4.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Top
| System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left));
this.panel4.AutoScroll = true;
this.panel4.AutoSize = true;
this.panel4.Controls.Add(this.testBlocks);
this.panel4.Controls.Add(this.chooseAllBlocks);
this.panel4.Location = new System.Drawing.Point(12, 261);
this.panel4.Name = "panel4";
this.panel4.Size = new System.Drawing.Size(392, 244);
this.panel4.TabIndex = 20;
//
// panel3
//
this.panel3.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.Bot
tom | System.Windows.Forms.AnchorStyles.Left));
this.panel3.AutoSizeMode =
System.Windows.Forms.AutoSizeMode.GrowAndShrink;
this.panel3.Controls.Add(this.startBtn);
this.panel3.Location = new System.Drawing.Point(12, 511);
this.panel3.Name = "panel3";
this.panel3.Size = new System.Drawing.Size(371, 90);
this.panel3.TabIndex = 18;
//
// allResult
//
this.allResult.Anchor =
((System.Windows.Forms.AnchorStyles)((System.Windows.Forms.AnchorStyles.T
op | System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
| System.Windows.Forms.AnchorStyles.Right));
```

## Продолжение приложения А

```

        this.allResult.AutoSizeColumnsMode =
System.Windows.Forms.DataGridViewAutoSizeColumnsMode.Fill;
        this.allResult.AutoSizeRowsMode =
System.Windows.Forms.DataGridViewAutoSizeRowsMode.AllCells;
        this.allResult.ColumnHeadersHeightSizeMode =
System.Windows.Forms.DataGridViewColumnHeadersHeightSizeMode.AutoSize;
        this.allResult.Columns.AddRange(new
System.Windows.Forms.DataGridViewColumn[] {
            this.test_datetime,
            this.test_name,
            this.test_description,
            this.test_succeed,
            this.test_res });
        dataGridViewCellStyle3.Alignment =
System.Windows.Forms.DataGridViewContentAlignment.MiddleLeft;
        dataGridViewCellStyle3.BackColor =
System.Drawing.SystemColors.Window;
        dataGridViewCellStyle3.Font = new System.Drawing.Font("Microsoft Sans
Serif", 8.25F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(204)));
        dataGridViewCellStyle3.ForeColor =
System.Drawing.SystemColors.ControlText;
        dataGridViewCellStyle3.SelectionBackColor =
System.Drawing.SystemColors.Highlight;
        dataGridViewCellStyle3.SelectionForeColor =
System.Drawing.SystemColors.HighlightText;
        dataGridViewCellStyle3.WrapMode =
System.Windows.Forms.DataGridViewTriState.True;
        this.allResult.DefaultCellStyle = dataGridViewCellStyle3;
        this.allResult.Location = new System.Drawing.Point(9, 14);
        this.allResult.Name = "allResult";
        this.allResult.RowHeadersVisible = false;
        this.allResult.Size = new System.Drawing.Size(573, 556);
        this.allResult.TabIndex = 17;
        //
        // test_datetime
        this.test_datetime.FillWeight = 49.86185F;
        this.test_datetime.HeaderText = "Время";
        this.test_datetime.Name = "test_datetime";
        //
        // test_name

```

## Продолжение приложения А

```
//
this.test_name.FillWeight = 97.85389F;
this.test_name.HeaderText = "Код теста";
this.test_name.Name = "test_name";
//
// test_description
//
this.test_description.FillWeight = 152.2843F;
this.test_description.HeaderText = "Название теста";
this.test_description.Name = "test_description";
//
// test_succeed
//
this.test_succeed.HeaderText = "";
this.test_succeed.Name = "test_succeed";
this.test_succeed.Visible = false;
//
// test_res
//
this.test_res.HeaderText = "Результат";
this.test_res.Name = "test_res";
this.test_res.Resizable =
System.Windows.Forms.DataGridViewTriState.True;
this.test_res.SortMode =
System.Windows.Forms.DataGridViewColumnSortMode.NotSortable;
this.test_res.Visible = false;
//
// panel5
//
this.panel5.AutoSize = true;
this.panel5.AutoSizeMode =
System.Windows.Forms.AutoSizeMode.GrowAndShrink;
this.panel5.Location = new System.Drawing.Point(383, 12);
this.panel5.Name = "panel5";
this.panel5.Size = new System.Drawing.Size(0, 0);
this.panel5.TabIndex = 21;
//
// panel6
this.panel6.Anchor =
((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.T
op | System.Windows.Forms.AnchorStyles.Bottom)
| System.Windows.Forms.AnchorStyles.Left)
```



## Продолжение приложения А

```
| System.Windows.Forms.AnchorStyles.Right));
this.panel6.Controls.Add(this.allResult);
this.panel6.Location = new System.Drawing.Point(383, 13);
this.panel6.Name = "panel6";
this.panel6.Size = new System.Drawing.Size(585, 588);
this.panel6.TabIndex = 18;
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.ClientSize = new System.Drawing.Size(989, 600);
this.Controls.Add(this.panel6);
this.Controls.Add(this.panel3);
this.Controls.Add(this.panel5);
this.Controls.Add(this.panel4);
this.Controls.Add(this.panel2);
this.Name = "Form1";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "АВТОТЕСТЫ";
this.panel1.ResumeLayout(false);
this.panel1.PerformLayout();
((System.ComponentModel.ISupportInitialize)(this.testBlocks)).EndInit();
this.panel2.ResumeLayout(false);
this.panel2.PerformLayout();
this.panel4.ResumeLayout(false);
this.panel4.PerformLayout();
this.panel3.ResumeLayout(false);
((System.ComponentModel.ISupportInitialize)(this.allResult)).EndInit();
this.panel6.ResumeLayout(false);
this.ResumeLayout(false);
this.PerformLayout();    }
#endregion
private System.Windows.Forms.Button startBtn;
private System.Windows.Forms.ComboBox serverList;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.TextBox mail;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.DateTimePicker date_begin;
private System.Windows.Forms.CheckBox currentPeriod;
```

## Продолжение приложения А

```
private System.Windows.Forms.CheckBox FizStart;
private System.Windows.Forms.CheckBox JurStart;
private System.Windows.Forms.CheckBox chooseAllBlocks;
private System.Windows.Forms.Panel panel1;
private System.Windows.Forms.DataGridView testBlocks;
private System.Windows.Forms.Panel panel2;
private System.Windows.Forms.Panel panel4;
private System.Windows.Forms.Panel panel3;
private System.Windows.Forms.Panel panel6;
private System.Windows.Forms.DataGridView allResult;
private System.Windows.Forms.Panel panel5;
private System.Windows.Forms.DataGridViewTextBoxColumn test_datetime;
private System.Windows.Forms.DataGridViewTextBoxColumn test_name;
private
    System.Windows.Forms.DataGridViewTextBoxColumn
test_description;
    private System.Windows.Forms.DataGridViewCheckBoxColumn test_succeed;
    private System.Windows.Forms.DataGridViewTextBoxColumn test_res;
}
}
```

## Продолжение приложения А

### Процедуры

#### Запуск автотестирования

```
use ISASHeat
```

```
go
```

```
if object_id('dbo.pTest_AllStart', 'p') is null
```

```
    exec ('create procedure dbo.pTest_AllStart
    as
```

```
        print "Заглушка" '      )
```

```
go
```

```
alter procedure dbo.pTest_AllStart
```

```
    @nvcRecipients nvarchar(200) = null,
```

```
    @dDateCalcBegin date = null,
```

```
    @bShowAllResults bit = 0,
```

```
    @bFizStart bit = 1,
```

```
    @bJurStart bit = 1,
```

```
    @bHelp bit = 0
```

```
/*
```

Название: pTest\_AllStart

Назначение: Пускач всех тестов

Входные данные:

@nvcRecipients nvarchar(200) = null, - почта для рассылки сообщений

@dDateCalcBegin date = null, - период для тестирования (начальная дата тестирования)

@bShowAllResults bit = 0, - показать результат выполнения

@bFizStart bit = 1, - запуск проверки по физ.лицам

@bJurStart bit = 1 - запуск проверки по юр.лицам

@bHelp - выводит блоки, на которые разделено тестирование

-- таблица с блоками для тестирования (может быть пустой или несозданной)

```
if object_id('tempdb.dbo.#ltHelp', 'u') is not null
```

```
    drop table #ltHelp
```

```
create table #ltHelp
```

```
(
```

```
    record_id          int not null,
```

```
    test_name          nvarchar(200) not null,
```

```
    test_description   nvarchar(500) null,
```

```
    is_checked         bit not null default(0),
```

```
    primary key(test_name)
```

```
)
```

## Продолжение приложения А

```
*/as

set nocount on
--#region Заполнение блоков для тестирования

if object_id('tempdb.dbo.#ltHelp', 'u') is null
    create table #ltHelp
    (
        record_id      int not null,
        test_name       nvarchar(200) not null,
        test_description nvarchar(500) null,
        is_checked      bit not null default(0),
        primary key(test_name)
    )
-- для запуска по блокам подсказка, какие блоки можно подать
if @bHelp = 1 or not exists (select null from #ltHelp)
begin
    truncate table #ltHelp
    insert #ltHelp
        (record_id, test_name, test_description, is_checked)
    values
        (1, N'TestReports', 'Проверка основных отчетов', 1),
        (2, N'TestBills', 'Проверка квитанций и счетов-фактур', 1),
        (3, N'TestOperations', 'Проверка наличия удаленных ссылок на операции', 1),
        (4, N'TestAccountActions', 'Проверка выполнения действий по ЛС', 1)
    if @bHelp = 1

    begin
        select * from #ltHelp
        return
    end
end
--#endregion Заполнение блоков для тестирования
--truncate table _tst_Results
if @bFizStart = 0 and @bJurStart = 0
begin
    raiserror('Не заданы данные для тестирования', 11, 1)
    return
end
```

## Продолжение приложения А

```
--#region Переменные
declare
    @iCalcPeriodRid int = (
        select
            cp.record_id
        from
            dbo.CalcPeriods as cp
        where
            cp.valid_end = '99991231' and
            cp.date_calc_begin = @dDateCalcBegin
    ),
    @iIndCalcPeriodRid int = (
        select
            icp.record_id
        from
            dbo.IndividualsCalcPeriods as icp
        where
            icp.valid_end = '99991231' and
            icp.date_calc_begin = @dDateCalcBegin
    ),
    @iBaseCalcPeriodRid int = (
        select
            bcp.record_id
        from
            dbo.BaseCalcPeriods as bcp
        where
            bcp.valid_end = '99991231' and
            bcp.date_calc_begin = @dDateCalcBegin
    )
--#endregion Переменные
--#region Юр.лица тестируем
if @bJurStart = 1
begin
    --#region Тестируем таблицу операций ЮЛ по объектам и договорам
    if exists(select null from #ltHelp where test_name = N'TestOperations' and
is_checked = 1)
    begin
        -- проверяем наличие удаленных ссылок на операции по объектам и
договорам
        exec dbo.pTest_ObjectsContractsOperationsCheckDeletedLinks
    end
end
end
```

## Продолжение приложения А

```
-- проверка использованного тарифа по операциям Начисление,  
Корректировка (должен быть тариф без НДС)  
exec pTest_ObjectsContractsOperationsCheckTarifValues  
    @iCalcPeriodRid = @iCalcPeriodRid  
  
-- проверка НДС в операциях по объектам договоров  
exec pTest_ObjectsContractsOperationsCheckNDSValues  
    @iCalcPeriodRid = @iCalcPeriodRid  
  
-- проверяем наличие дважды снятых начислений по объектам и договорам  
exec pTest_ObjectsContractsOperationsCheckDoubleDeleted  
end  
--#endregion Тестируем таблицу операций ЮЛ по объектам и договорам  
--#region Основные отчеты Юр.лиц (Итоговый, По срокам задолженности),  
Журнал регистрации сф)  
if exists(select null from #ltHelp where test_name = N'TestReports' and is_checked  
= 1)  
begin  
    /*****  
    * сводим Итоговый ЮЛ и По срокам задолженности с/ф ЮЛ  
    *****/  
    -- заполнение отчета Итогового по каждой группе услуг за  
текущий(выбранный) месяц  
    exec dbo.pTest_MonthTotalForm  
        @iCalcPeriodRid = @iCalcPeriodRid,  
        @bSelectedPeriodPrev = 0  
    -- заполнение отчета Итогового по каждой группе услуг за прошлый месяц  
    exec dbo.pTest_MonthTotalForm  
        @iCalcPeriodRid = @iCalcPeriodRid,  
        @bSelectedPeriodPrev = 1  
    -- заполнение отчета по срокам задолженности за текущий месяц  
    exec dbo.pTest_DebtsPeriodForm  
        @iCalcPeriodRid = @iCalcPeriodRid  
    -- проверки по отчетам Итоговый и По срокам задолженности  
    exec dbo.pTest_MonthTotalCheckSumAndCount  
    exec dbo.pTest_MonthTotalCheckNDSSum  
    exec dbo.pTest_MonthTotalCheckSaldo  
    exec dbo.pTest_MonthTotalCheckSelectedAndPrev
```

## Продолжение приложения А

```
exec dbo.pTest_MonthTotalDebtsPeriodsChecksum
exec dbo.pTest_DebtsPeriodCheckTotalSum
/*****
* сводим Итоговый ЮЛ и Журнал с/ф ЮЛ
*****/
-- заполнение отчета Журнал регистрации счетов-фактур
exec dbo.pTest_ReportInvoicesForm
    @iCalcPeriodRid = @iCalcPeriodRid
-- заполнение отчета Итогового за текущий(выбранный) период по
бригадам не ЧД
exec dbo.pTest_MonthTotalForReportInvoicesForm
    @iCalcPeriodRid = @iCalcPeriodRid,
    @bSelectedPeriodPrev = 0
-- сверка суммы в отчете Итоговом и отчете Журнал регистрации
exec dbo.pTest_MonthTotalReportInvoicesChecksum
end
--#endregion Основные отчеты Юр.лиц (Итоговый, По срокам
задолженности), Журнал регистрации сф)

--#region Проверки счетов-фактур
if exists(select null from #ltHelp where test_name = N'TestBills' and is_checked =
1)
begin
-- проверки НДС
exec dbo.pTest_InvoicesNDSCheck
    @iCalcPeriodRid = @iCalcPeriodRid
-- Проверка кол-во*тариф = сумма
exec dbo.pTest_InvoicesQuantitySumsCheck
    @iCalcPeriodRid = @iCalcPeriodRid
end
--#endregion Проверки счетов-фактур
end
--#endregion Юр.лица тестируем
--#region Физ.лица тестируем
if @bFizStart = 1
begin
--#region Тестируем наличие удаленных операций по ЛС (после 02.2014)
if exists(select null from #ltHelp where test_name = N'TestOperations' and
is_checked = 1)
begin
-- проверяем наличие удаленных ссылок на операции по ЛС
exec dbo.pTest_AccountsOperationsCheckDeletedLinks
```

## Продолжение приложения А

```
end
--#endregion Тестируем наличие удаленных операций по ЛС (после 02.2014)

--#region Квитанции ЛС ФЛ
if exists(select null from #ltHelp where test_name = N'TestBills' and is_checked =
1)
begin
-- проверки
exec dbo.pTest_AspAccountsBillsCheckSumAndCount
    @iBaseCalcPeriodRid = @iBaseCalcPeriodRid
exec dbo.pTest_AspAccountsBillsCheckSumTotal
    @iBaseCalcPeriodRid = @iBaseCalcPeriodRid
exec dbo.pTest_AspAccountsBillsCheckIndicationsDiff
    @iBaseCalcPeriodRid = @iBaseCalcPeriodRid
end
--#endregion Квитанции ЛС ФЛ
--#region Основные отчеты физ.лиц (Итоговый, По срокам задолженности)
if exists(select null from #ltHelp where test_name = N'TestReports' and is_checked
= 1)
begin
/*****
* сводим Итоговый ФЛ с отчетом по срокам задолженности
*****/
-- заполняем итоговый за выбранный месяц
exec dbo.pTest_IndMonthTotalFlatForm
    @iCalcPeriodRid = @iIndCalcPeriodRid,
    @bSelectedPeriodPrev = 0
-- заполняем итоговый за прошлый месяц
exec dbo.pTest_IndMonthTotalFlatForm
    @iCalcPeriodRid = @iIndCalcPeriodRid,
    @bSelectedPeriodPrev = 1
-- заполняем Итоговый с тарифами
exec dbo.pTest_IndMonthTotalDetailedForm
    @iCalcPeriodRid = @iIndCalcPeriodRid
-- заполняем отчет по срокам задолженности
exec dbo.pTest_IndDebtsPeriodsForm
    @iCalcPeriodRid = @iIndCalcPeriodRid
-- проверки по отчету Итоговому
exec dbo.pTest_IndMonthTotalCheckSaldo
exec dbo.pTest_IndMonthTotalCheckSelectedAndPrev
exec dbo.pTest_IndMonthTotalCheckDetailed
exec dbo.pTest_IndMonthTotalCheckSumAndCount
exec dbo.pTest_IndMonthTotalDebtsPeriodsCheckSum
```



## Продолжение приложения А

```
exec dbo.pTest_IndDebtsPeriodCheckTotalSum
end
--#endregion Основные отчеты физ.лиц

--#region Выполнение различных действий по ЛС (подключение, отключение,
обследование...)
if exists(select null from #ltHelp where test_name = N'TestAccountActions' and
is_checked = 1)
begin
-- проверка периода подключения по услугам

exec dbo.pTest_AccountConnectActCheckConnectionPeriod
-- должна быть возможность ввода 2 обследований подряд
exec dbo.pTest_AccountConnectActCheckTwoInspections
-- не должно быть возможности ввода 2 подключений подряд (если один из
них не подключение по оплате абон.задолженности)
exec dbo.pTest_AccountConnectActCheckTwoConnections
-- не должно быть возможности ввода 2 отключений подряд (если один из
них не отключение по абон.задолженности)
exec dbo.pTest_AccountConnectActCheckTwoDisconnections
-- TODO
-- проверка ввода подключения по оплате абон.задолженности без
отключения по задолженности
-- проверка ввода подключения по оплате абон.задолженности и еще 1 вида
подключения после 2 отключений, корректность определения периода
отключения
end
--#endregion Выполнение различных действий по ЛС (подключение,
отключение, обследование...)
end
--#endregion Физ.лица тестируем
--#region Проверка результатов и отправка на почту

/*****

* отправка результатов

*****/

-- запускаем проверку по всем выполненным тестам
```

## Продолжение приложения А

```
-- отправляем результат на почту

--declare @nvcRecipients nvarchar(500) = N'osindeyeva@alseco.kz'
exec dbo.pTest_CheckResults
    @nvcRecipients = @nvcRecipients,
    @dDate = null

if @bShowAllResults = 1
    select * from dbo._tst_Results order by test_datetime desc
--#endregion Проверка результатов и отправка на почту
go
--exec dbo.pTest_AllStart
declare
    @procname    nvarchar(255) = 'pTest_AllStart',
    @comment     nvarchar(255) = 'Пускач всех тестов'
if exists(
    select
        null
    from
        sys.objects o
    inner join sys.extended_properties ep_o
        on
            ep_o.major_id = o.object_id and
            ep_o.minor_id = 0 and
            ep_o.name = 'MS_Description'
    where
        o.name = @procname
)

execute sp_dropextendedproperty 'MS_Description',
    'user',
    'dbo',
    'procedure',
    @procname

execute sp_addextendedproperty 'MS_Description',
    @comment,
    'user',
    'dbo',
    'procedure', @procname go
```