

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**
Ордена Трудового Красного Знамени
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский технический университет связи и информатики»

Кафедра «Информатика»

Курсовая работа

по дисциплине «Алгоритмы и алгоритмические языки» на тему:

Эксплуатация корпоративных информационных систем

Выполнил: студент группы БФИ-2202

Сидорук Д. В.

Научный руководитель: старший

преподаватель кафедры

«Информатика» Загвоздкина А. В.

Москва, 2023 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
ТЕХНИЧЕСКОЕ ЗАДАНИЕ.....	7
1.1 Введение.....	7
1.2 Основание для разработки.....	7
1.3 Назначение разработки.....	7
1.4 Требования к программе или программному изделию.....	7
1.4.1 Требования к функциональным характеристикам.....	7
1.4.2 Требования к надежности.....	8
1.4.3 Условия эксплуатации.....	8
1.4.4 Требования к составу и параметрам технических средств.....	8
1.4.5 Требования к информационной и программной совместимости.....	9
1.4.6 Требования к маркировке и упаковке.....	9
1.5 Требования к транспортированию и хранению.....	10
1.6 Требования к программной документации.....	10
1.7 Технико-экономические показатели.....	11
1.8 Стадии и этапы разработки.....	11
1.9 Порядок контроля и приемки.....	11
ГЛАВА 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ.....	12
1.1 Развитие корпоративных информационных систем.....	12
1.2 Эксплуатация и сопровождение информационных систем.....	25
1.3 Разработка и внедрение корпоративных информационных систем предприятия.....	34

ГЛАВА 2 ПРАКТИЧЕСКАЯ ЧАСТЬ.....	40
2.1 Разработка электронного конспекта.....	41
2.2 Разработка тестовой программы.....	42
2.3 Дополнительные возможности программы.....	61
ЗАКЛЮЧЕНИЕ.....	74
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	75
ПРИЛОЖЕНИЕ А.....	76
ПРИЛОЖЕНИЕ В.....	78

ВВЕДЕНИЕ

Актуальность эксплуатации корпоративных информационных систем обусловлена рядом факторов, которые определяют неотложность решения сложных задач, связанных с оптимизацией ресурсов, улучшением качества услуг и повышением конкурентоспособности организаций на рынке. В современном мире информационные системы являются неотъемлемой частью функционирования любой компании, и их эффективное использование становится ключевым фактором успеха.

Прежде всего, актуальность данной темы обусловлена быстрым развитием информационных технологий и постоянным увеличением объемов данных, которые необходимо хранить, обрабатывать и анализировать. В условиях глобализации и цифровизации экономики, корпоративные информационные системы становятся основой для принятия обоснованных управленческих решений, а также инструментом для оптимизации бизнес-процессов и повышения операционной эффективности.

Во-вторых, актуальность темы подчеркивается необходимостью снижения издержек и рационального использования ресурсов в условиях жесткой конкуренции на рынке. Корпоративные информационные системы позволяют автоматизировать рутинные операции, сокращать время на выполнение задач, уменьшать вероятность ошибок и повышать прозрачность бизнес-процессов. **Таким образом**, компании, активно использующие информационные системы, получают значительное конкурентное преимущество перед своими соперниками.

В-третьих, актуальность темы связана с постоянным ростом требований к качеству услуг, предоставляемых организациями. Эффективная эксплуатация корпоративных информационных систем позволяет предоставлять клиентам

более высокий уровень сервиса, оперативно реагировать на их запросы и предложения, а также адаптироваться к изменяющимся условиям рынка.

Кроме того, актуальность темы "Эксплуатация корпоративных информационных систем" обусловлена необходимостью обеспечения информационной безопасности в организациях. С учетом постоянного роста киберугроз и уязвимостей информационных систем, важность их надежной защиты и своевременного обнаружения инцидентов безопасности становится все более очевидной. Компетентное управление информационными системами и их защитой является одним из ключевых факторов успешного функционирования организаций в современных условиях.

Также актуальность темы подкрепляется необходимостью интеграции различных информационных систем в единую корпоративную среду. Интеграция позволяет обеспечить более эффективное взаимодействие между различными подразделениями компаний, повысить мобильность и доступность информации, а также облегчить процесс принятия управленческих решений.

В заключение, актуальность эксплуатации корпоративных информационных систем определяется неотложностью решения сложных задач, связанных с оптимизацией ресурсов, улучшением качества услуг и повышением конкурентоспособности организаций на рынке. Эффективное использование информационных систем способствует развитию науки и техники в данной области, а также обеспечивает стабильное функционирование и успешное развитие организаций в условиях постоянно меняющегося мира.

Объект исследования: Эксплуатация информационных систем на предприятиях.

Предмет исследования: Использование программного обеспечения для автоматизации бизнес-процессов в компаниях.

Цель работы: исследование актуальных требований к выпускникам специальностей, связанных с экономикой и управлением организациями и разработка программного продукта, являющегося программой для обучения и тестирования студентов по дисциплине «Эксплуатация корпоративных информационных систем».

Задачи:

1. Изучение и анализ знаний, умений и навыков, которые необходимы для работы в сфере управления организациями;
2. Разработка обучающей программы, содержащей теоретический материал и тест.

Методы исследования: Для решения поставленных задач были использованы теоретические методы исследования. Теоретическую основу исследования составили труды в области эксплуатации корпоративных информационных систем И. С. Гантц, Е. Е. Майорова и И. С. Таюрской

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1.1 Введение

Областью применение разрабатываемого программного продукта является углубленное изучение, а также контроль качества усвоения студентами основных определений, принципов и особенностей, используемых в сфере эксплуатации корпоративных информационных систем.

1.2 Основание для разработки

- Лист утверждения тем курсовых работ, подписанный руководителем: доцентом кафедры «Информатика» Московского технического университета связи и информатики Гуриковым С. Р;
- Наименование разработки – электронное пособие по теме: эксплуатация корпоративных информационных систем.

1.3 Назначение разработки

Разрабатываемый программный продукт предназначен для изучения и, в последующем, контроля качества усвоения студентами основных определений, принципов и методов, используемых в сфере эксплуатации корпоративных информационных систем.

1.4 Требования к программе или программному изделию

1.4.1 Требования к функциональным характеристикам

Программная разработка должна быть написана в четком соответствии с материалами занятий по дисциплине «Алгоритмы и алгоритмические языки».

Структуру программного кода тестовых вопросов изменять нельзя, она должна соответствовать материалам занятий;

Студенту, за счет часов самостоятельной работы, разрешается заниматься дополнительным поиском информации с целью расширения возможностей своей программной разработки, с последующим описанием их в пояснительной записке.

Необходимо в четком соответствии с материалами лекций и практических работ, состоявшихся в семестре, реализовать следующие индивидуальные задачи:

- создание меню с вложенными пунктами для чтения информации из текстового файла, содержащего теоретическую часть;
- создание строки состояния с выводом текущей даты и времени, а также версии программы;
- создание медиаплеера, поддерживающего просмотр (воспроизведение) фото-, аудио- и видео- файлов;
- реализовать возможность вывода результатов записи ответов на тест, представленных в виде двумерного массива, в компонент DataGridView, осуществления вставки столбца из значения, заданного пользователем (запрос организовать на основе функции InputBox), записи результирующей матрицы в документ Microsoft Word;
- упорядочить результаты теста, представленные в виде одномерного массива в другом компоненте DataGridView, с использованием метода сортировки простым обменом (1 способ), при этом используется алгоритм сортировки, показанный на занятиях (изменять его нельзя). Осуществить запись результатов теста в Microsoft Excel и выполнить построение графика (диаграммы) на рабочем листе. Кроме формирования рабочей книги и графика

(диаграммы), программа выполняет фотографию графика, записывая ее по указанному пользователем пути. Осуществить нахождение суммы значений полученных результатов тестирования, используя встроенную функцию SUM() в программном коде метода;

- осуществить вывод результатов тестирования в Microsoft Excel и осуществить программным кодом построение диаграммы, отображающей результаты тестирования.

Кроме того необходимо:

- с целью упрощения отладки и сопровождения программы, реализовать режим разработчика, позволяющий переходить на форму с результатами тестирования без необходимости прохождения тестирования;
- создать индикатор, отображающий прогресс прохождения тестирования;
- реализовать индикатор, отображающий количество оставшихся для решения вопросов;
- обеспечить возможность вывода упорядоченных с использованием метода сортировки простым обменом (1 способ) результатов тестирования в документ Microsoft Excel и выполнения построения графика (диаграммы) на рабочем листе, выполнения фотографии графика (диаграммы) и записи его по указанному пользователем пути;
- обеспечить возможность открытия документов Microsoft Word и Microsoft Excel с результатами тестирования;
- создать установщик программы.

1.4.2 Требования к надежности

Разрабатываемое программное обеспечение должно быть спроектировано таким образом, чтобы обеспечить защиту и надежную работу при наличии ошибок во входных данных и/или от некорректных действий пользователя – предполагается, что программный продукт должен быть спроектирован таким образом, чтобы внутренняя или внешняя (некритическая для системы) ошибка не приводила к аварийной остановке.

1.4.3 Условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявленным к техническим средствам в части условий их эксплуатации.

Программа будет работать в температурном режиме от +5 до +35 °C при относительной влажности 90% и атмосферном давлении 462 мм. РТ. Ст., поскольку такие условия приблизительно соответствуют условиям эксплуатации современных компьютеров непромышленного исполнения.

1.4.4 Требования к составу и параметрам технических средств

Для корректной работы программного продукта вычислительная система должна обладать следующими характеристиками:

- процессор с тактовой частотой не менее 1 ГГц;
- оперативная память объемом не менее 4ГБ ОЗУ;
- периферийные устройства: клавиатура, мышь;
- монитор с минимальным разрешением 1280x1024;
- 500МБ памяти на жестком диске.

Для корректной работы вычислительной среды необходимо наличие системного программного обеспечения, основным элементом которого является операционная система. В связи с тем, что Microsoft является лидером в сегменте продаж операционных систем, то целесообразно использовать операционную систему семейства Windows не ниже Windows 8.

1.4.5 Требования к информационной и программной совместимости

Требования к информационным структурам (файлов) на входе и выходе не предъявляются.

Исходные коды программы должны быть реализованы на языке C# с использованием Windows Forms. В качестве интегрированной среды разработки программы должна быть использована среда Microsoft Visual Studio.

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы.

1.4.6 Требования к маркировке и упаковке

Программа поставляется в виде программного изделия на внешнем флеш-носителе.

Программное изделие должно иметь маркировку с обозначением товарного знака компании-разработчика, типа (наименования), номера версии, порядкового номера, даты изготовления и номера сертификата соответствия Госстандарта России (если таковой имеется). Маркировка должна быть нанесена на программное изделие в виде наклейки, выполненной полиграфическим способом с учетом требований ГОСТ 9181-74.

Упаковка флеш-носителя – пакет для хранения.

Упаковка программного изделия должна проводиться в закрытых вентилируемых помещениях при температуре от плюс 15 до плюс 40 °С и относительной влажности не более 80% при отсутствии агрессивных примесей в окружающей среде.

Подготовленные к упаковке программные изделия укладываются в тару, представляющую собой коробки из картона. Для заполнения свободного пространства в упаковочную тару укладываются прокладки из гофрированного картона или пенопласта. На верхний слой прокладочного материала укладывается товаросопроводительная документация – упаковочный лист и ведомость упаковки.

1.5 Требования к транспортированию и хранению

- транспортировка разрабатываемого программного продукта должна осуществляться студентом в назначенный срок по указанию преподавателя, принимающего результаты курсовой работы;
- ответственным за хранение программного продукта является студент;
- результаты выполненной работы хранятся на кафедре «Информатика» в течение года.

1.6 Требования к программной документации

Предварительный перечень программной документации:

- пояснительная записка к курсовой работе, оформленная в соответствии с ГОСТ 7.32-2001;

1.7 Технико-экономические показатели

В данной работе не рассчитываются.

1.8 Стадии и этапы разработки

Стадии и этапы разработки определены в план-графике выполнения курсовой работы.

1.9 Порядок контроля и приемки

- контроль и приемка программного продукта осуществляется в течение семестра поэтапно в соответствии с план-графиком;
- курсовая работа подлежит защите, в ходе которой студент представляет свою разработку;
- в случае невыполнения план-графика, студенты могут быть выданы дополнительные задания;
- в случае регулярного невыполнения план-графика, студенту может быть выдано задание на защиту курсовой работы, в ходе выполнения которого он обязан продемонстрировать свои практические навыки;
- оценку программного продукта производят преподаватели осуществляющие проверку курсовых работ.

ГЛАВА 1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Развитие корпоративных информационных систем

Развитие корпоративных информационных систем (далее — КИС) можно описать как последовательность выхода стандартов информационных систем, которые определяют сложность и уровень задач, решаемых с помощью информационных систем на предприятиях. Стандарты описывают структуру и требования к корпоративным информационным системам предприятия и набор правил к реализации информационных систем (далее — ИС) на предприятиях. На рисунке 1 выделены основные вехи развития КИС. На текущем этапе рекомендуемыми к КИС остаются требования ориентирования на клиента и подчинения требованиям ведения электронного бизнеса. С этим связан прочий набор требований и варианты реализации КИС. Впрочем, информационные системы, реализованные в соответствии с любым описанным ниже стандартом, также относят к классу корпоративных информационных систем.

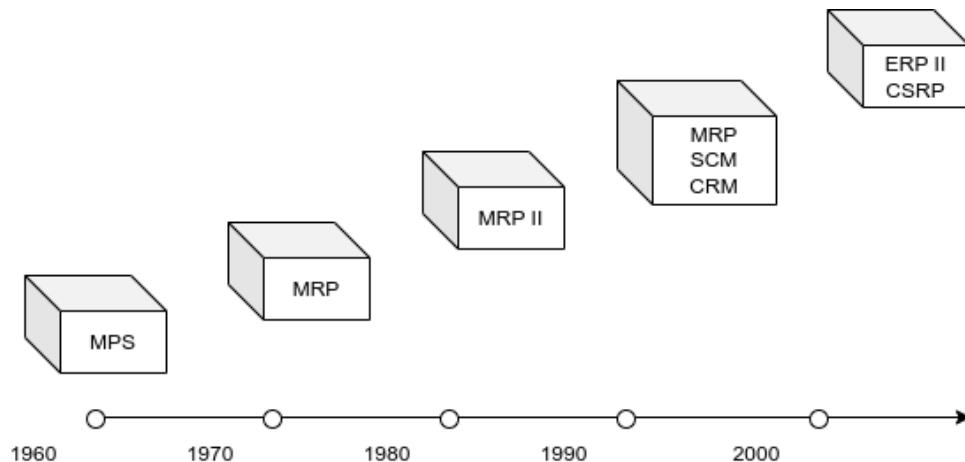


Рисунок 1 — История развития КИС

Стандарт MPS (Master Planning Scheduling)

Управление календарным планированием

MPS = планирование производства и закупок

Основой для формирования корпоративных информационных систем стал стандарт, который начал развиваться еще в шестидесятых годах 20 века. Основная функция стандарта Управления календарным планированием – это составление основного плана производства. Согласно данному подходу план выпуска готовой продукции формировался на основе данных спроса.

Основными предпосылками создания данного подхода являлись задержки в производстве по причине запаздывания поставок комплектующих и сырья. Также важный момент, который сказался на необходимости развития данного подхода, это затоваривание складов сырьем, которое поступало раньше намеченного срока. Таким образом, объемно календарное планирование – это основная составляющая стандарта. Здесь сначала формируется укрупненный план продаж, затем осуществляется его распределение по периодам, далее на его основе формируется план-график закупки для собственного производства (см. Рисунок 2). Такой подход и сейчас является основой работы небольших торговых предприятий. Основная сложность – это прогнозирование объема и сроков поставок, что требует планирования потребительского спроса, длительности производства и складских платежей. В рамках стандарта используются такие понятия как Точка заказа и Уровень пополнения. Точка заказа – это уровень складских запасов, при снижении которого необходимо сделать заказ поставщику. Уровень пополнения запаса товара на складе – это количество товара, выше которого не рекомендуется повышаться уровень складского запаса.

Стандарт MRP (Material Requirement Planning)

Планирование материальных потребностей

MRP = MPS + выполнение Производства и Закупок + учет сроков поставки

Следующий подход, который фактически является основой для современных существующих корпоративных информационных систем, это Планирование материальных потребностей. Согласно данному стандарту определяется процесс планирования материальных ресурсов, автоматически предусматривается создание проектов заказа на закупку сырья для внутреннего производства. Основной принцип состоит в следующем — все материалы и комплектующие должны поступать в производство в запланированное время для того, чтобы готовый продукт можно было получить без дополнительных задержек. Таким образом, данный стандарт ускоряет доставку тех материалов, которые необходимы для производства в первую очередь. Это позволяет избежать ситуации, когда задержка поставки одного материала блокирует выпуск продукции даже при наличии всех прочих комплектующих. Таким образом, планирование, контроль и обработка дат поступления материалов позволяет снизить складские издержки.

Компьютерные технологии в рамках данного стандарта используются для решения задач управления производственными запасами. Сам подход поддерживает два основных принципа «Just in time»: «Заказ вовремя» и «Вовремя извести». Данный подход и сейчас используется на предприятиях среднего масштаба с серийным типом производства. Наиболее важное понятие, которое вводится в обиход в рамках данного стандарта , это «Bill of material» (BOM), спецификация изделий, которая показывает структуру изделия с точки зрения сырья, комплектующих, полуфабрикатов, включает учет времени

доставки и планов выпуска готовой продукции. Фактически БОМ представляет собой структуру конечного продукта, при этом спецификация связана с производственным заказом, выдача которого может привести к резервированию компонентов по спецификации. Если данных компонентов нет на складе, то формируются заявки на поставку.



Рисунок 2 — Стандарт MRP.

Планирование материальных
потребностей

Стандарт Планирование материальных потребностей следует принципу зависимого спроса, то есть если есть потребность в конечном изделии, то есть потребность во всех компонентах, необходимых для производства. Также здесь используется принцип обеспечения компонентами как можно позже, чтобы уровень запасов был минимален, но достаточен для производства. Для учета временной зависимости производственных процессов информационной

системе необходимо знать объем выпуска продукции, логическую цепочку, структуру изделий, а также продолжительность технологических операций. На основании плана выпуска, спецификации изделий и самой технологической цепочки в системе осуществляется расчет потребности в материалах и зависимость от конкретных сроков выполнения технологических операций. Серьезным недостатком данного подхода является то, что не учитывается загрузка оборудования, амортизация производственных площадей, стоимость рабочей силы и потребляемой энергии.

Основной производственный план, как правило, формируется для пополнения запасов готовой продукции и удовлетворения заказов потребителей. На практике формируется черновой вариант плана для оценки возможности обеспечения реализации материальными ресурсами и мощностями. Затем осуществляется детализация в разрезе материальных составляющих. Если необходимая номенклатура и количественный состав сырья и комплектующих не присутствует в запасе или недостаточен для производства, то план должен быть соответствующим образом скорректирован. После того, как основной производственный план утвержден как действующий, на его основе осуществляется запуск производственных заказов. Сама ведомость материалов представляет собой таблицу базы данных (далее — БД), которая отражает данные состава изделия. Состояние таблицы корректируется. Текущее состояние запасов материалов отображается в соответствующих таблицах БД с указанием всех необходимых характеристик материалов.

Стандарт CRP (Capacity Requirements Planning)

Управление загрузкой производственных мощностей

Перечисленные проблемы стали стимулом для развития следующего класса информационных систем. Стандарт Управление производственными

мощностями с учетом ресурсных ограничений производств стал следующим шагом развития. Результатом работы данного стандарта стало определение производственных мощностей, которые представлялись как набор ресурсов, человеческих, рабочих групп, технических и самого оборудования. В случае, если мощностей недостаточно для покрытия требований плана MRP, уменьшается потребность в продукции либо увеличивается производительность и численность человеческих или технических ресурсов. Соответственно для определения потребности в возможностях в качестве исходных данных выступают: календарный план, доступные производственные мощности, а также технологические карты, которые связывают потребность производства компании в затрачиваемых мощностях в соответствии с конкретным технологическим процессом. Уровень сложности решаемых здесь задач значительно выше, чем в MRP. Станки могут переналаживаться, использоваться для различных операций, кроме того, на станках может работать один рабочий или используются графики сменности, можно учитывать перерывы, перенастройки, переналадки и прочее. Большинство простых систем данного класса максимально могут лишь учитывать сменность. Для решения прочих вопросов используется понятие задел производственной мощности, таким образом сглаживается появление текущих проблем.

Планирование производственных мощностей в соответствии с данным стандартом осуществляется по каждому виду продукции, которая включена в календарный план. В рамках этой системы осуществляется балансировка загрузки рабочих центров для обеспечения выполнения календарного плана. Если выявляется расхождение в загрузке мощностей с учетом ограничений, то выдается предупреждающее сообщение и проводятся действия по корректировке операций, распределение нагрузки на другие рабочие центры

либо на другие производственные программы (см. Рисунок 3). При этом функцию оптимизации загрузки рабочих центров оставляют человеку.



Рисунок 3 — Управление загрузкой производственных мощностей

Производственная программа должна соответствовать реальной загрузке рабочих центров, производственных мощностей. Здесь производственная программа становится основой для определения потребности в ресурсах.

Недостатком данных систем является ограниченный учет производственных факторов и отсутствие моделей, которые могут использоваться для оптимизации загрузки рабочих центров.

Системы класса DRP (Distribution Requirements Planning)

Автоматизированные системы планирования потребностей распределения

Следующий класс Автоматизированные системы планирования потребностей распределения. Системы данного класса применялись в сфере продаж. Основные задачи, решаемые в рамках систем — это контроль запасов в распределенной сети, координация спроса и предложения, отслеживание поставок от производства до продажи, которая рассматривается как основная функция. Основой реализации данной системы является производственное расписание, которое определяет пополнение запасов в распределенной торговой системе. Система выполняет агрегирование данных по прогнозам, о фактически поступивших заказах, планирование производства с указанием конкретных дат, расчет количества комплектующих изделий и формирование плана готовой продукции. Основная функция системы планирования потребностей распределения — это планирование транспортных перевозок. На основе заявки на транспортное обслуживание в режиме реального времени составляются и редактируются графики перевозок и долгосрочные планы работы для расчета потребности в транспортных средствах.

MRP замкнутого цикла

MRP замкнутого цикла = MRP + отслеживание всех отклонений от плана

Следующий этап развития КИС — развитие стандарта MRP Замкнутого цикла. Основная особенность данного подхода - непрерывное отслеживание выполнения плана снабжения и производственных операций. В функционал информационных систем были добавлены возможности аналитического контроля произведенной готовой продукции и затраченных комплектующих. Соответственно здесь же отслеживались задержки производственных заказов, объемы, динамика продаж и прочее. В результате анализа исходной информации предлагаются различные корректировки программы производства и портфеля заказов. Доработка данной системы позволяет включить

дополнительные функции планирования — это планирование производства и планирование мощности. В системе также реализованы обратные связи, которые позволяют отслеживать текущее состояние производственной системы. В результате получается замкнутый цикл управления и планирования ресурсов.

Стандарт MRP II (Manufacturing Resource Planning)

Планирование производственных ресурсов

MRP II = MRP замкнутого цикла + CRP + планирование и выполнение Продаж

Следующая веха — важный этап развития КИС — это стандарт MRP II, который расшифровывается как Планирование производственных ресурсов. Если в стандарте MRP планирование ограничено потребностью материалов, то здесь уже обсуждается планирование производственных ресурсов предприятия. Система данного класса предназначена для прогнозирования, планирования и контроля производства, начиная от закупки сырья и заканчивая продажей готовой продукции клиенту. В состав систем данного класса входят блоки планирования продаж и производства, планирования потребности в сырье, комплектующих, план потребности в мощностях, выполнение плана производства с использованием мощностей и закупки материалов. Кроме этого налаживаются обратные связи между всеми указанными элементами.

Блок планирования продаж в данном стандарте позволяет оценить объем и динамику клиентских продаж, которые необходимы для достижения стратегических задач компании. Модуль планирования производства определяет план производства продукции в разрезе ассортиментных групп. В качестве основы расчета лежит план продаж. Планирование производственной

мощности позволяет преобразовать план производства в конкретные единицы загрузки рабочих центров, то есть станков и прочего оборудования с учетом загрузки рабочих. Модуль планирования потребности в сырье и материалах позволяет сформировать расписание закупок и внутреннего производства. Все документы формируются на основе согласования с другими. Блок выполнения производства и пополнения комплектующих включает исполнение, контроль и отслеживание производственной деятельности предприятия, а обратная связь позволяет решать все основные проблемы со снабжением сырьем для выпуска готовой продукции, а также корректировать планы, которые невыполнимы. В качестве ресурсов здесь рассматриваются людские ресурсы, материалы и денежные средства.

Стандарт ERP (Enterprise Resource Planning)

Планирование ресурсов предприятия

ERP = MRPII + реализация всех типов производства + интегрирование планирования ресурсов (Финансовых, Человеческих и Складских) по различным направлениям деятельности компании + многозвенное планирование

Следующий этап развития КИС — это развитие класса систем ERP Планирование ресурсов предприятия. Системы данного класса охватывают финансовую, хозяйственную, производственную и прочую деятельность компаний. К основным требованиям реализации относят клиент-серверную архитектуру и принцип открытых систем. Системы создавались для представления руководству информации, которая способствует принятию управленческих решений, а также с целью информационного обеспечения электронного обмена данными с поставщиками и клиентами, а также между подразделениями организации.

Цель разработки корпоративных информационных систем данного класса — это объединение в единое информационное пространство управления материальными, финансовыми, человеческими ресурсами, закупками, сбытом, управление запасами, складом, а также бухгалтерский и налоговый учет, расчеты с покупателями и поставщиками. В едином хранилище данных хранится информация о предприятии, которая накапливается в процессе ведения деловой, финансовой, производственной и кадровой деятельности.

От типичной MRP системы ERP отличается техническими характеристиками, а именно графическим интерфейсом пользователя, использованием реляционной базы данных, архитектуры клиент-сервер и использованием принципа переносимости на основе работы открытых систем. Как правило, ERP-система строится по модульному принципу, модули охватывают ключевые процессы деятельности организации.

К основным функциями ERP-систем относят ведение конструкторских технологических спецификаций, которые определяют состав изделий, а также материальные ресурсы операции, необходимые для изготовления, учитываемые при формировании планов продаж производства, планировании потребности в материалах и комплектующих, а также при расчете объемов поставок для выполнения плана продукции, управления запасами и закупками, ведения договоров, реализации централизованных закупок, обеспечения учета, оптимизации складских и цеховых запасов, включая планирование производственных мощностей от укрупненного планирования до использования отдельных станков оборудования. Оперативное управление финансами включает составление финансового плана, осуществление контроля, финансовый и управленческий учет, управление проектами, в том числе планирование этапов, ресурсов, необходимых для реализации.

ERP-система — информационная система для идентификации и планирования всех ресурсов предприятия, которые необходимы для осуществления продаж, производства, закупок и учета в процессе выполнения клиентских заказов.

Методология ERP – это методология эффективного планирования и управления всеми ресурсами предприятия, которые необходимы для осуществления продаж, производства, закупок и учета при исполнении заказов клиентов в сферах производства, дистрибуции и оказания услуг

Методология CSRP (Customer Synchronized Resource Planning)

Планирование ресурсов вместе с клиентами

$$\text{CSRP} = \text{ERP} + \text{EAI} + \text{CRM}$$

CRM (Customer Relationship Management) – Управление взаимоотношениями с клиентами

EAI (Enterprise Application Integration) – Интеграционное программное обеспечение

Следующим этапом развития КИС является оптимизация управления ресурсами предприятия с помощью стандарта CSRP Планирование ресурсов вместе с клиентами. Данный стандарт обеспечивает интеграцию приложений управления взаимоотношениями с клиентами класса CRM с базовым функционалом ERP-систем. Для внедрения данного стандарта необходимо оптимизировать производственную деятельность компании, интегрировать информацию по работе с покупателями с работой планирующих и производственных подразделений, а также добавить возможность использования открытых технологий, поддержки интеграции покупателей,

поставщиков и системы управления производством. Таким образом , стандарт перераспределяет акцент ведения бизнеса, то есть работа фокусируется на рыночной активности, деятельности по управлению взаимоотношениями с покупателями и убирается с производственной деятельности предприятия.

В центре поддержки покупателей становятся управление продажами и поддержка пользователей, то есть обработка заказов и одновременное продвижение новых и сопутствующих продуктов и услуг. Сама система планирования производства и деятельности предприятия становится более динамичной. Напрямую организуется взаимодействие модуля планирования и конфигуратора продукции, то есть в режиме реального времени предоставляется возможность рассчитать различные варианты производственных графиков, оценить предполагаемые затраты на ресурсы и на переналадку оборудования, динамично скорректировать. Ориентир данных систем на клиентах позволяет организовать также персональное обслуживание, что дает возможности ясно видеть потребности покупателей.

ERP II (Enterprise Resource and Relationship Processing)

Управление предприятием на основе ресурсов и взаимоотношений

Компоненты ERP II:

SRM (Supplier Relationship Management) – система управления взаимоотношениями с поставщиками

CRM (Customer Relationship Management) – система управления связями с клиентами

SCM (Supply Chain Management) – система управления виртуальными логистическими цепочками

BI (Business Intelligence) – система бизнес-аналитики

PLM (Product Lifecycle Management) – система управления жизненным циклом продукта

HRM (Human Resource) – система управления «человеческими ресурсами»

Financials – система управления финансами со стороны различных участников

Mobile Business – мобильный бизнес

KM (Knowledge Management) – система управления знаниями о бизнесе

Следующим этапом развития или расширения функций КИС является класс ERP II, Управление предприятием на основе ресурсов и взаимоотношений. Данный класс объединяет в себя ранее созданные стандарты, то есть планирование ресурсов с клиентами с учетом управления жизненным циклом продукта, взаимоотношений с поставщиками и посредниками и ряд других функций. Стандарт в настоящее время продолжает развиваться.

Основное направление работы данных систем – управление информацией компании не только для внутренних целей, но и для развития отношений сотрудничества с другими организациями или физическими лицами. Здесь используется интернет-ориентированная архитектура, которая существенно отличается от архитектуры традиционных ERP-систем. Используется работа с веб-клиентами, распределенными технологиями обработки, часто применяются облачные технологии.

1.2 Эксплуатация и сопровождение информационных систем

В данном разделе эксплуатация и сопровождение корпоративных информационных систем будет рассматриваться с точки зрения управления ИТ сервисами. Вспомним, что такое ИТ-сервис и чем он интересен, а также какими характеристиками принято его описывать.

ИТ-сервис в корпоративной среде — это ИТ-услуга, которую ИТ департамент оказывает или предоставляет бизнес подразделениям предприятия. В качестве примера ИТ-сервиса можно привести специальное программное обеспечение, модули, которые используют специалисты отдельных предприятий, типовые ИТ сервисы, которыми пользуются все специалисты предприятия, электронная почта, системы хранения данных и проч.

ИТ-сервис характеризуется рядом параметров.

Функциональность включает в себя функции, которые решает сервис, это задачи, которые определяются предметной областью использования данного сервиса. Таким образом, функциональность привязана к конкретному бизнес-процессу или бизнес-функции, исполняемой специалистом.

Время обслуживания — это период времени, в течение которого ИТ подразделение поддерживает данный ИТ-сервис, то есть несёт ответственность за его непрерывное функционирование. Как правило, на предприятиях используются режимы обслуживания 24*7 или 8*5.

Следующий параметр, который характеризует ИТ сервис, это доступность — доля согласованного времени обслуживания, в течение которого ИТ-сервис доступен для пользователей. Здесь учитывается время на простой. Например, если мы указываем доступность 95 %, то при режиме 8 * 5 = 40 часов в неделю сервис будет простоявать 5% = 2 часа в неделю.

Надёжность определяется средним временем наработки на отказ, то есть средним временем между двумя сбоями. Соответственно , если мы говорим о том, что, доступность ИТ-сервиса составляет 95%, следовательно, сервис 2 часа в неделю не будет работать. Если известно, что два раза в неделю сервис будет недоступен, то можно рассчитать надежность — время непрерывной работы- среднее время между двумя сбоями ($42 / 2 - 1 = 19$ часов).

Параметр производительность это способность информационной системы соответствовать предъявленным к ней требованиям и выполнять поставленные задачи, измеряется в количестве операций за определенное время.

Следующий важный параметр особенно интересен с точки зрения КИС конфиденциальность, то есть вероятность несанкционированного доступа к данным или несанкционированного изменения данных. Количественные метрики оценки параметра обычно не используются, конфиденциальность ИТ-сервиса определяется внутренними распоряжениями на предприятии. Если данные сервиса относятся к высокому классу секретности, то даже элементарной доступ на просмотр считается опасным.

Параметр масштаб характеризует объём и сложность работ по поддержке ИТ-сервиса.

Затраты — стоимость всех ресурсов, которые вовлечены в сопровождение ИТ-сервиса, учитываются затраты от простоев ИТ-сервис, заработка сотрудников, стоимость внешних услуг и прочие расходы.

Подход управления ИТ на основе услуг (IT Service Management) основан на использовании знаний библиотеки ITIL. Библиотека ITIL — это библиотека лучших практик, которую собрали в конце 1980х по заказу Британского правительства. ITIL — это передовая практика управления ИТ-услугами,

которая используется во многих организациях по всему миру. Целая философия ITIL вокруг книг-руководств включает поддерживающую схему сертификации и квалификации. Идеей разработки ITIL является признание того, что организации становятся все более зависимыми от ИТ, чтобы удовлетворить свои корпоративные цели и удовлетворить свои бизнес-потребности. Это приводит к повышенным требованиям к надежным, высококачественным ИТ - услугам.

ITIL обеспечивает основу для качественного управления ИТ-услугами посредством задокументированных, проверенных процессов, которые охватывают весь Жизненный цикл службы.

ITIL Версия 3 была официально выпущена 5 июня 2007 года. ITIL соответствует различным международным международный стандарт ISO/IEC 20000 управления).

С точки зрения поддержки ИТ-сервисов в рамках подхода ITIL принято описывать основные процессы: управление инцидентами, управление проблемами, управление конфигурациями, управление изменениями и управление релизами.

Рассмотрим их кратко.

Любое отклонение, любое событие, которое отличается от прописанных норм в договоре, от нормального состояния считается **инцидентом**. Инцидент — любое событие, которое не является частью стандартных операций предоставления услуги.

Сам процесс решения инцидента выглядит следующим образом. Сотрудник сервисной службы регистрирует в журнале инцидент, получив заявку от пользователя, определяет ему приоритет, классифицирует его и при

возможности осуществляет начальную поддержку — предлагает перезагрузить компьютер, выключить и включить устройство, провести откат транзакции. Служба Service Desk отличается от службы Help Desk уровнем начальной поддержки, если в функции Help Desk входит только регистрация и классификация инцидентов с учетом приоритетов, то Service Desk берет на себя некоторые работы по элементарной поддержке. После выполнения начальной поддержки специалист Service Desk принимает решение, требуется ли специализированное обслуживание по данному инциденту. Если не требуется, то инцидент закрывается на данном уровне, при необходимости дополнительной поддержки передается на следующий уровень, то есть для решения инцидентов привлекаются специализированные подразделения, оперативный персонал, который уже с помощью базы знаний и собственных навыков пытается решить данный инцидент. Решение инцидентов сопровождается выездом или выходом специалиста на место возникновения инцидента, проведением манипуляций и использованием уже известной базы решений подобных вопросов. Если на данном уровне инцидент не был решен, то осуществляется эскалация на следующий уровень, то есть привлекаются специалисты более высокого уровня поддержки.

Рассмотрим следующий процесс поддержки ИТ-сервисов — процесс **управления проблемами**. Процесс управления проблемами предназначен для минимизации негативного влияния инцидентов на бизнес и уменьшение количества инцидентов. Проблемой считается инцидент или набор инцидентов, связанные с одной причиной, причем причина которых неизвестна. Фактически проблема — это причина возникновения инцидентов.

При реализации процесса выполняют следующие функции: анализ инцидентов, регистрация проблем, идентификация корневых причин

инцидентов, выявление известных ошибок, управление известными ошибками, решение проблемы и закрытие проблемы. Проблема считается закрытой, если выявлена известная ошибка и установлен порядок устранения. Цель блока управления проблемами — это установление корневых причин ряда инцидентов для их предотвращения.

Процесс поддержки ИТ-сервисов **управления конфигурациями** используется для помощи управления инфраструктурой оборудования. Основное понятие, которое используется в рамках процесса конфигурационная единица (CI — Configuration Item) — представляет собой элемент или сущность, которая описывается набором атрибутов. В зависимости от правил принятых на предприятии к конфигурационной единице может относиться как оборудование, так и программные средства, файлы, документация, персонал. Оборудование или программное средство описываются детально либо укрупненно, например, рабочее место может описываться как стационарный компьютер, АРМ, а может описываться элементами монитор, клавиатура, компьютерная мышь и так далее. В зависимости от выбранной детализации осуществляется процесс управления конфигурациями, то есть при замене конфигурационной единицы должна выполняться вся процедура документального оформления.

Процесс **управления изменениями** связан со всеми вышеперечисленными процессами, то есть любое решение, которое принимается в рамках процессов управления ИТ-сервисами, как правило, регистрируется в процессе управления изменениями. Данный блок важен для планирования и согласования любых управленческих воздействий, которые влияют на работу специалистов, на функционал бизнес-процессов. Процесс предполагает регистрацию всех существенных изменений в среде

информационных систем предприятия, включает правила оформления разрешений на внесение изменений, разработку графика изменений, организует взаимодействие ресурсов и оценивает воздействие любого изменения на информационные системы предприятия с учетом рисков.

Для управления изменениями создаются специальные органы, которые принимают решения о необходимости проведения рассматриваемого вида изменений. Как правило, это комитет по управлению изменениями, включает ИТ-специалистов, начальников соответствующих отделов, а также руководство предприятия. Основная цель работы комитета — провести только те изменения, которые принесут пользу и будут реализованы имеющимися ресурсами.

Процесс управления изменениями действует как регулятор, который позволяет гибко, но стablyно работать с основным процессом управления ИТ-сервисами. Корректирующие меры позволяют уменьшить количество инцидентов. Сфера управления процессом аналогична сфере управления в блоке управления конфигурациями. То есть если принято, что-то к элементам конфигурации относят только крупные блоки, то изменения регистрируют только воздействия, связанные с данными блоками полностью. Для того, чтобы провести такое изменение, оформляется специальный запрос.

Процесс управления релизами предназначен для согласования изменений, которые вносятся в ИТ-инфраструктуру предприятия. Релиз — это набор новых или изменённых позиций конфигурации, которые тестируются и внедряются совместно. Сам процесс состоит из трёх этапов : разработка, тестирование и внедрение. Этап разработки часто может выполняться с помощью аутсорсинга. На этапе тестирования определяются критерии, которые позволяют оценить, готов релиз к внедрению или не готов. После подготовки релиза сама реализация осуществляется с помощью запроса на изменение.

Основные функции процесса управления релизами — это планирование, проектирование, подготовка релиза, обучение пользователей, аудит оборудования до начала внедрений, размещение копий ПО, установка и совершенствование оборудования и постоянное улучшение процесса.

Блок управления конфигурациями, как правило, реализуется при управлении крупными проектами, которые сопровождаются аудитом инфраструктуры, то есть для того, чтобы описать конфигурационные единицы, используемые на предприятии, необходимо понять цель реализации.

Описание ИТ-инфраструктуры предприятия или описание системы управления конфигурациями осуществляется чаще всего в двух случаях — для поддержки работы блоков управления ИТ-сервисами и при реализации ИТ-проектов. Для описания единиц конфигурации проводится аудит инфраструктуры, который включает следующие работы: выявление проблем функционирования с определением рекомендаций, оценку качества оборудования, оценку стоимости конфигурационных единиц, оценку стоимости владения оборудованием, документирование информации об инфраструктуре предприятия и подготовку эксплуатационной документации. На первом этапе сбора данных определяются границы работ, рабочие места, оборудование и цель проведения аудита. Сбор данных включает в себя этап интервьюирования персонала, приборные измерения, сбор оперативной информации и осмотр оборудования. В ходе интервьюирования проводится опрос специалистов, использующих ПО или оборудование, опрос специалистов, обслуживающих и отвечающих за работоспособность. Осмотр оборудования зависит от цели исследования.

Процесс управления доступностью. Данный процесс обеспечивает разработку, изменение и оптимизацию ИТ-сервисов для достижения

оптимального согласованного уровня доступности. В рамках данного процесса проходит проектирование систем, разрабатываются планы обслуживания, обеспечения безопасности и проводятся прочие работы. Основная цель процесса — оптимизировать возможности инфраструктуры сервисов и ИТ-подразделения для предоставления эффективного с точки зрения затрат и постоянного уровня доступности, который позволяет бизнесу работать в эффективном режиме и достигать своих целей.

Доступность в данном случае — это способность ИТ-сервисов выполнять функции в установленный момент или за установленный период времени. Доступность подкрепляется надежностью, восстанавливаемостью ИТ-инфраструктуры, а также эффективностью работы организации. В рамках данного процесса реализуют следующие функции: инвентаризация ресурсов ИТ, анализ доступности ИТ-сервисов, в том числе оборудования, программного обеспечения, каналов связи, определение узких мест ИТ-сервисов с точки зрения доступности, регистрация проблем доступности, которые угрожают невыполнением SLA и подготовка рекомендаций по их устранению.

Процесс управление непрерывностью предоставления ИТ-сервисов обеспечивает подготовку к чрезвычайным ситуациям, включает планирование поведения сотрудников при возникновении проблем, инцидентов, оценивает существующие информационные системы. Основная цель данного процесса — обеспечить восстановление технических средств, всей инфраструктуры поддержки услуг при возникновении необходимости.

Понятие устойчивость используется для обозначения способности ИТ-службы и ИТ-инфраструктуры поддерживать сервис в работоспособном состоянии в случае столкновения с чрезвычайной ситуацией. Основная цель данного процесса поддержка непрерывности бизнеса. На время возникновения

ситуации предоставление ИТ-услуг должно поддерживаться на аварийном уровне, то есть уровне приемлемом для ведения бизнеса, минимально необходимым для его функционирования.

Процесс предоставления ИТ-сервисов **управление финансами** иногда в литературе обозначается как процесс управления затратами. Процесс обеспечивает возможность учета финансовых факторов при поддержке и разработке ИТ-сервисов. Основная цель данного процесса — обеспечить эффективное с точки зрения управления затратами управление ИТ-активами, которые используются при оказании ИТ услуг.

В рамках процесса отслеживаются фактические затраты в разрезе заказчиков, ИТ-сервисов, пользователей, рассчитываются внутренние цены на ИТ услуги, формируется информация о стоимости предоставляемых ИТ сервисов, собирается информация о клиентах. Здесь также реализуются следующие функции: прогноз затрат и выручки, разработка бюджета сервисов, калькулирование счёта, предоставление его бизнес-пользователям, анализ использования сервиса и связанных с этими издержек, поиск путей снижения издержек и разработка системы выставления счетов за услуги. Процесс предоставляет бизнесу и ИТ количественную и финансовую оценку активов, которые лежат в основе ИТ-услуг. Как правило, процесс управления финансами включает в себя три основных функции: составление бюджета, бухгалтерский учет и выставление счетов.

1.3 Разработка и внедрение корпоративных информационных систем предприятия

При разработке корпоративных информационных систем особенности деятельности влияют как на структуру информационной системы, так и на

этапы внедрения, разработки или доработки программных средств. В то же время структуры различных предприятий в целом похожи между собой, независимо от вида деятельности предприятия состоят из ряда подразделений. Таким образом, любую организацию можно представить в виде взаимодействующих элементов (подразделений), каждый из которых имеет свою собственную сложную структуру и набор функционала. В общем случае предприятие описывается с помощью функциональных связей, информационных и внешних связей.

Поскольку предприятия обладают общими признаками, процесс разработки корпоративных информационных систем формулируется как набор единых принципов. Причем сам процесс разработки КИС рассматривается с различных точек зрения. Первый подход представляет собой описание набора функционала или набора минимальных требований, которые должна выполнять КИС, включая последовательность действий, потоков работ, исполнителей, информационные взаимосвязи, алгоритмы обработки информации. Второй подход представляет описание стадий жизненного цикла разработки системы, набор этапов, иначе говоря цикл процесса разработки. Во всех случаях разработку (или доработку КИС, что чаще случается) рассматривают как ИТ-проект.

Любой ИТ-проект обладает рядом свойств, которые не зависят от предметной области. Сам проект можно описать с помощью характеристик, которые отличают его от любого другого объекта управления:

- Изменчивость. Цель любого проекта - перевод системы из текущего состояния в желаемое, целевое.

- Ограниченностю конечной цели. Цель должна быть конкретной, определенной и понятной. Цель описывается в соответствующих предметной области терминах.
- Ограничение продолжительности. Если не будет указан срок действия проекта, то любая разработка может затянуться до невероятных сроков.
- Ограниченностю бюджета и ограниченность ресурсов. Все виды ресурсов, выделяемых на проект, конечны, имеют ограничение.
- Новизна для предприятия. Проект должен быть важен и нов для предприятия.
- Комплексность. Большое число факторов, которые прямо или косвенно влияют на процесс и результаты проекта.
- Правовое и организационное обеспечение. На время реализации проекта создается специфическая организационная структура, набирается команда проекта, организуется взаимосвязь между членами команды и определяется набор функций, ответственности.

Эффективность работ по проекту достигается за счет распределения ресурсов, координации выполняемой деятельности и управления процессами реализации проекта. Таким образом, можно сказать, что проект описывается как некий блок, где на входе определены технические требования, условия финансирования, а на выходе четко описан результат проекта, достижение которого означает эффективное завершение. Контроль выполнения стадий проекта проводится по выделенным параметрам.

Когда мы говорим о разработке или доработке КИС, то можно выделить основные фазы проектирования ИС (см. Рисунок 4):

- формирование концепции, определение требований;
- разработка технического задания;
- проектирование или разработка технического проекта;
- изготовление (разработка, реализация), подготовка рабочей документации;
- внедрение и ввод системы в эксплуатацию.

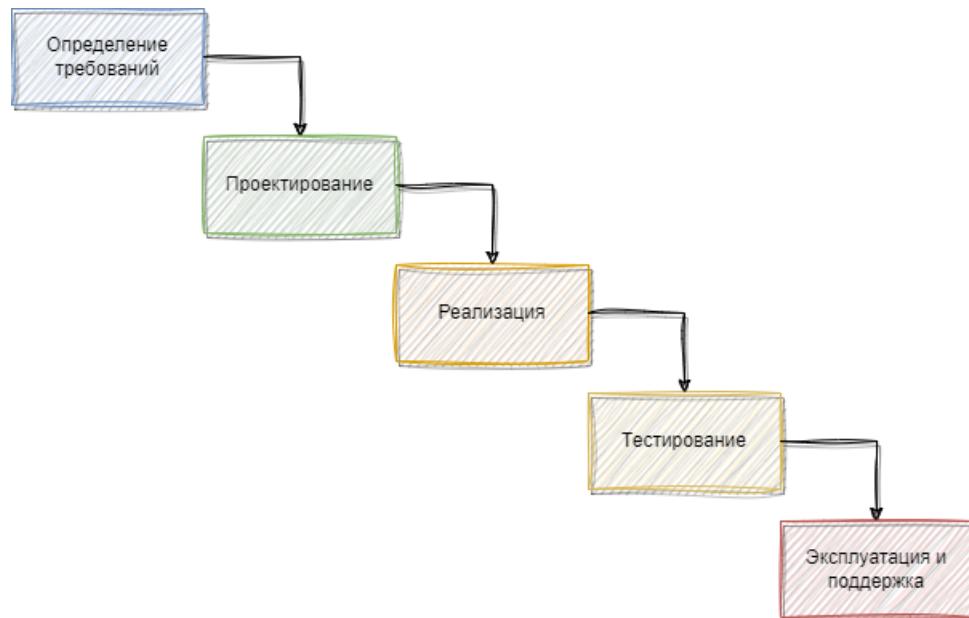


Рисунок 4 — Жизненный цикл информационных систем

Концептуальная фаза включает в себя следующие работы: формирование идеи, постановку целей, формирование обоснованного предложения заказчику о создании КИС, определяются основные функции, технические характеристики, формируется ключевая команда проекта. Основные выходные документы стадии – это технико-экономическое обоснование целесообразности создания КИС с выбранными функциями и характеристиками, анализ мотивации и требований заказчика и других участников, заявка на создание ИС, исходные технические требования, проверка данных на соответствие ГОСТам и прочим стандартам. Здесь же определяются основные ограничения проекта, в том числе материальные, финансовые, трудовые. На данном этапе также проводится сбор исходных данных и анализ текущего состояния, сравнительная оценка альтернатив, представление предложений, их экспертиза и утверждение.

Основной целью стадии разработки технического предложения является подтверждение целесообразности создания ИС. На данном этапе описываются функции и технические характеристики, проектируется совокупность всех работ, в том числе НИОКР, проектных, монтажно-наладочных, определяются сроки выполнения, составляются сметы и бюджет проекта, определяются исполнители, в том числе внешние организации, описываются материалы, которые необходимы для проведения проектных работ. В результате на выходе данного этапа формируются такие документы как техническое предложение на создание ИС, научно-технический отчет, который содержит результаты предпроектных исследований и эскизный проект ИС. Сам документ технического предложения определяет цели, требования, исходные данные, необходимые для разработки ИС. На данном этапе проводится декомпозиция модели проекта, то есть расписывается более детально подсистемы, функциональные задачи. Разрабатываются требования к подсистемам, к

математическому и программному обеспечению, комплексу программных и технических средств, которые будут использоваться в рамках ИС, определяется перечень задач, составляется календарный график работ, определяются этапы создания системы, сроки выполнения, проводится предварительный расчет затрат на создание системы, определяется уровень экономической эффективности внедрения.

Основная задача этапа проектирования – разработка технических решений по созданию системы и окончательное определение ее стоимости. Здесь формируется документ технический проект, в котором уже определены функции ИС, этапы с начальными и конечными датами, конкретные требования по срокам, стоимости и ресурсам, определяется объем работ, сроки выполнения, себестоимость, экономическая эффективность, которая обеспечивается реализацией проекта, описывается социальная и общественная значимость проекта. Комплект документов включает конкретные технические задания, специфические инструкции, спецификации и соответствующие документы по системе.

Разработка завершается выполнением технических работ по подготовке ПО. Здесь оформляется рабочая документация на ИС, данный документ должен быть достаточен для приобретения, монтажа и наладки комплекса технических средств, которые необходимы для работы системы, включает документацию по аппаратному обеспечению и программному обеспечению для установки, а также эффективной эксплуатации системы. Состав и содержание документации определяется и корректируется заказчиком в зависимости от степени детализации решения, описанного на предыдущем этапе в проектной документации. На этапе разработки также осуществляется тестирование, контролирование функций системы, оценка основных показателей проекта.

Внедрение или ввод системы в эксплуатацию – это передача действующей системы для установки на предприятии. Здесь собирается информация о качестве созданной системы, работоспособности, реальной эффективности функционирования системы, проводится анализ функционирования, надежности, рассчитываются экономические показатели эффективности, оценивается функционально-алгоритмическая полнота системы. Данные по анализу работы системы накапливаются, используются для последующих проектов, проводится анализ состояния и определяются дальнейшие направления развития. На данном этапе готовится и подписывается документация, осуществляется сервисное обслуживание и реализуются программы сопровождения.

ГЛАВА 2 ПРАКТИЧЕСКАЯ ЧАСТЬ

Программа для курсовой работы была разработана в среде программирования Microsoft Visual Studio 2022 и реализована на языке программирования Visual C#.

При запуске программы пользователь переходит на форму заставки приложения. На форме представлены следующие элементы (см. Рисунок 5):

- label, позволяющий отобразить на экране основную информацию о курсовой работе;
- button, позволяющий перейти на другие формы или выйти из программы;
- flowLayoutPanel, не видимый визуально, но позволяющий выровнять кнопки ровно по центру экрана.

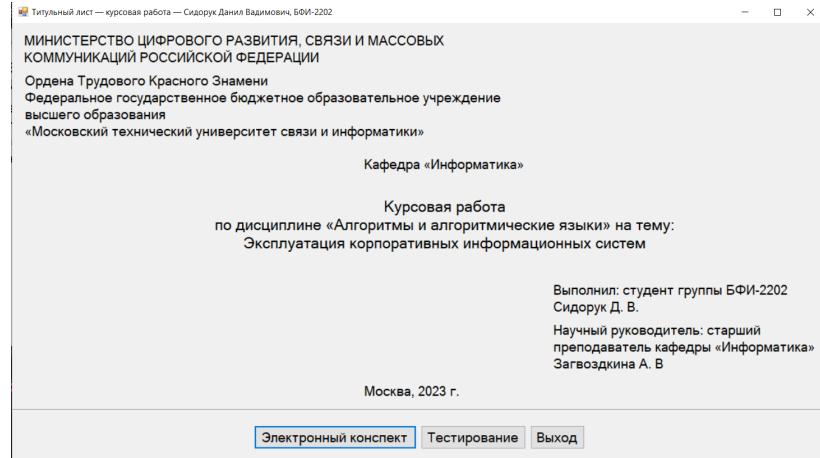


Рисунок 5 — Заставка программы

2.1 Разработка электронного конспекта

Для того, чтобы разработать электронный конспект, понадобилось создать форму, в которой был отражен теоретический материал по вышеупомянутой теме. На форме представлены следующие элементы (см. Рисунок 6):

- button, позволяющий перейти на другие формы или выйти из программы;
- webView2, служащий для отображения pdf файла с теоретической информацией.

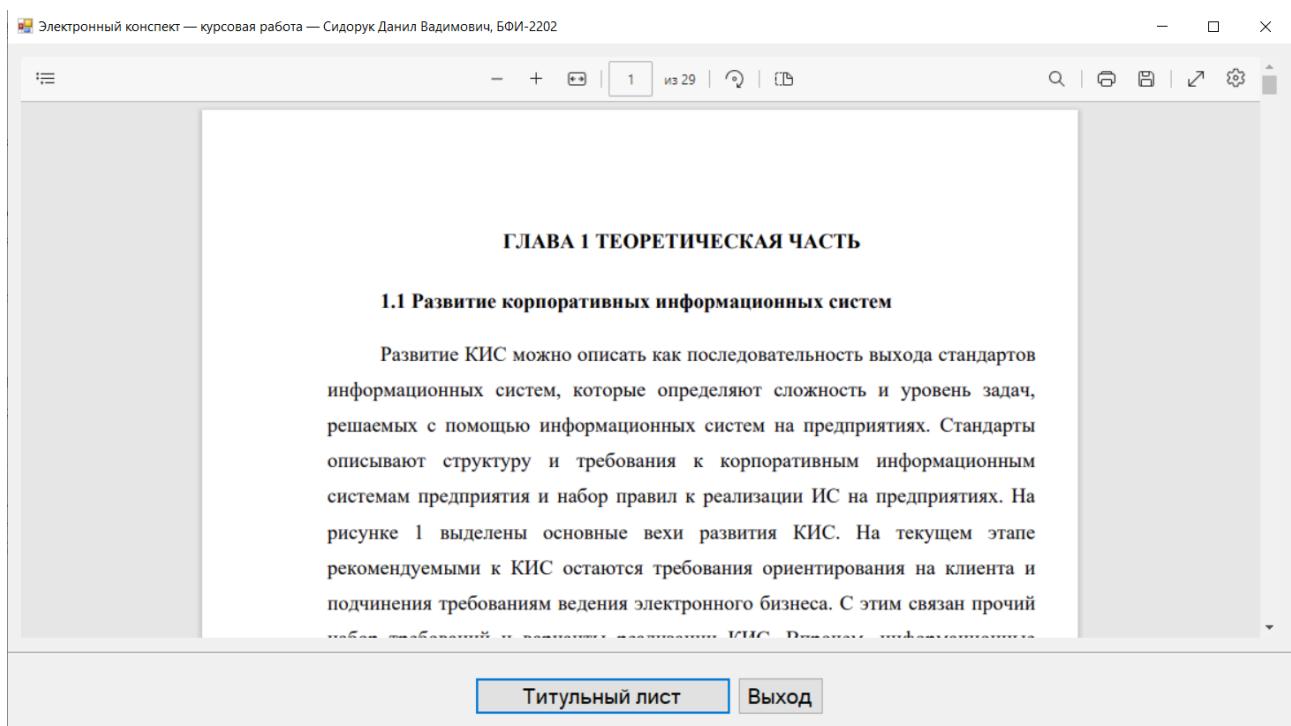


Рисунок 6 — Электронный конспект

2.2 Разработка тестовой программы

Для теста были разработаны 16 вопросов с использованием 8 компонентов ввода. Вопросы теста представлены в приложении А. Для доступа к каждому следующему вопросу пользователю необходимо ответить на текущий вопрос, следуя указаниям в задании, и нажать кнопку «Подтвердить». Если пользователь не дал никакого ответа, программа выдает сообщение об этом. Если пользователь ответил на вопрос неверно, программа выдает сообщение о неправильном ответе. После правильного ответа на вопрос пользователь получает сообщение о правильном ответе. Как в случае правильного, так и в случае неправильного ответа, после вывода сообщения об ответе, правильный он или неправильный, происходит переход на форму следующего вопроса.

На рисунке 7 продемонстрирован вид всех возможных сообщений о пользовательском вводе.

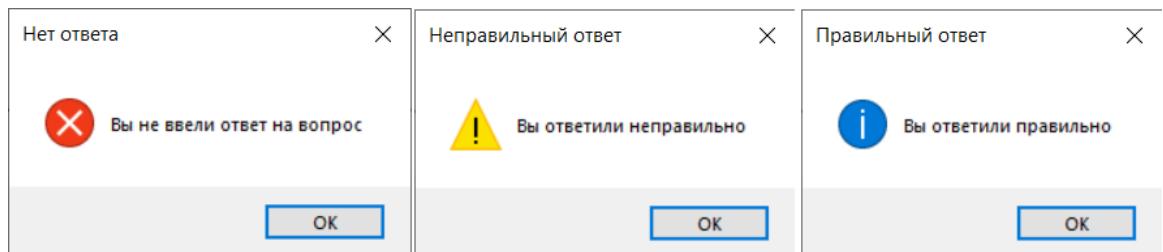


Рисунок 7 — Различные виды сообщений о пользовательском вводе

Перейдем к описанию использованных компонентов

TextBox – поле ввода текста с клавиатуры, был использован для вопросов, касающихся определений, которые обучающийся должен помнить наизусть.

На рисунке 8 и рисунке 9 продемонстрированы формы вопросов, содержащие данный компонент.

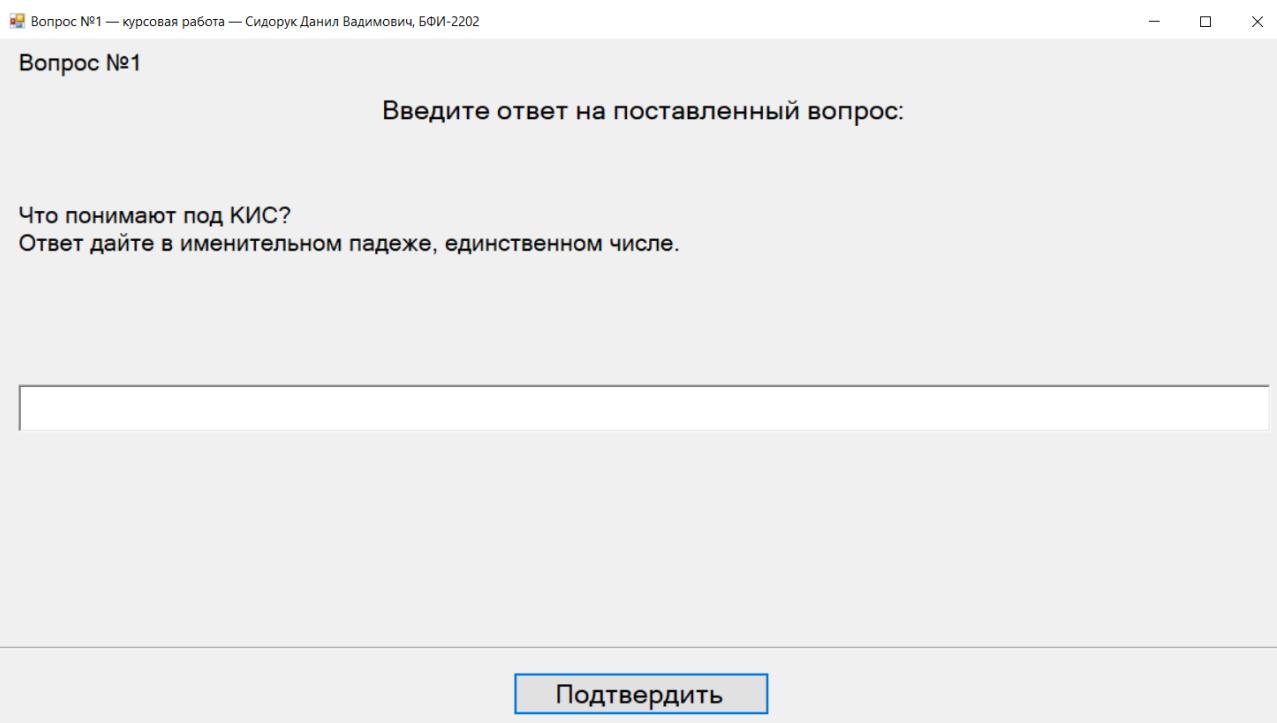


Рисунок 8 — Первый вопрос, содержащий компонент TextBox

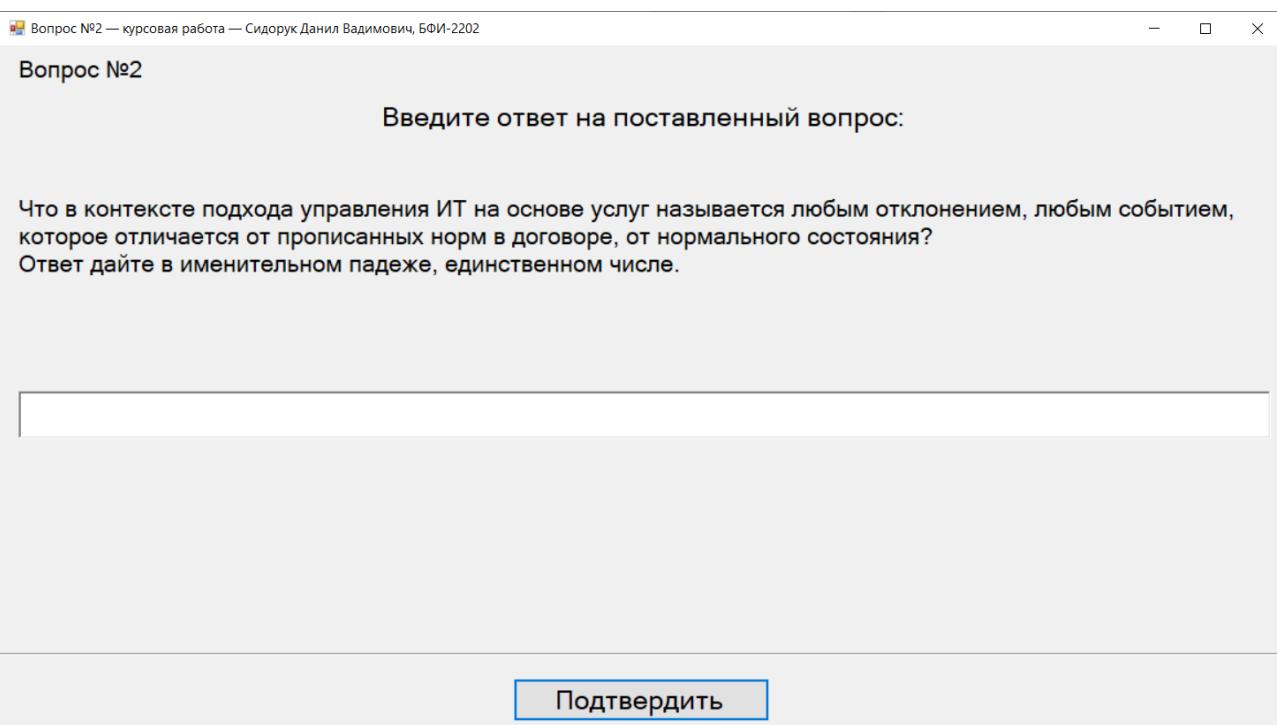


Рисунок 9 — Второй вопрос, содержащий компонент TextBox

`RadioButton` – переключатель. Если хранить несколько элементов управления этого типа в одном контейнере, то при выборе одного элемента все остальные тут же перестанут быть выбранными, поэтому этот компонент очень удобен для выбора одного из нескольких вариантов ответа.

На рисунке 10 и рисунке 11 продемонстрированы формы вопросов, содержащие данный компонент.

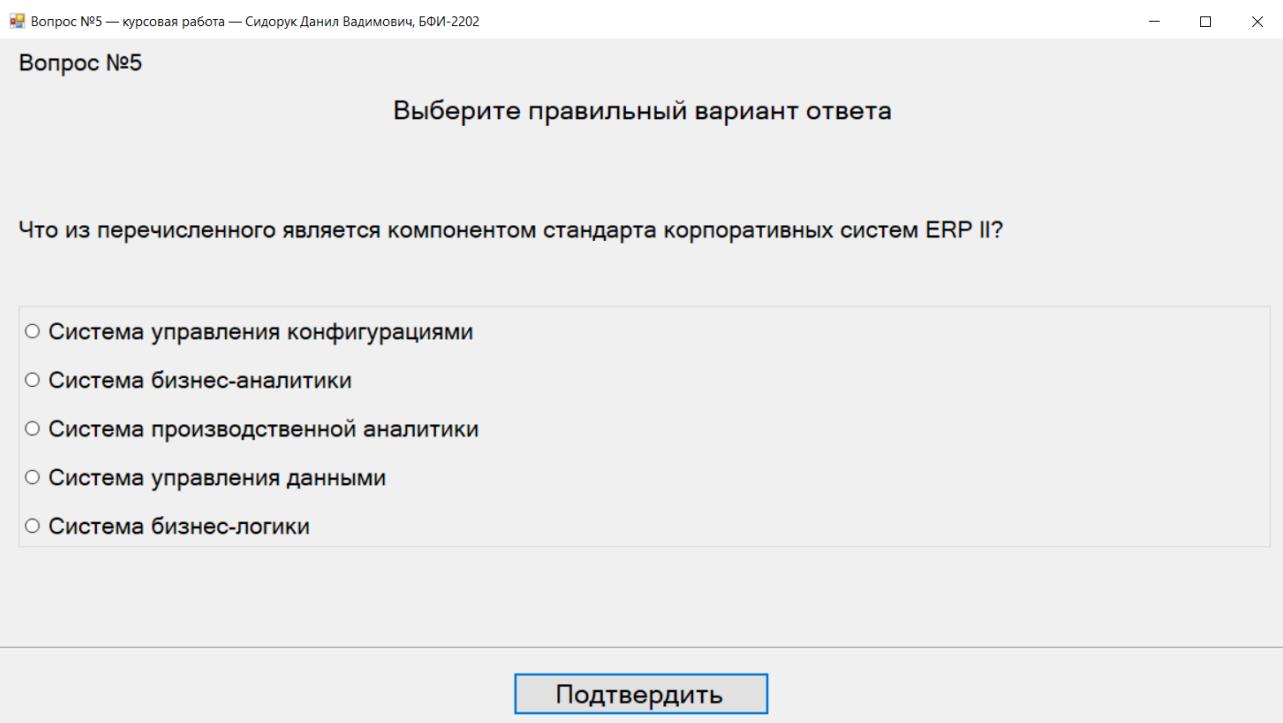


Рисунок 10 — Первый вопрос, содержащий компонент RadioButton

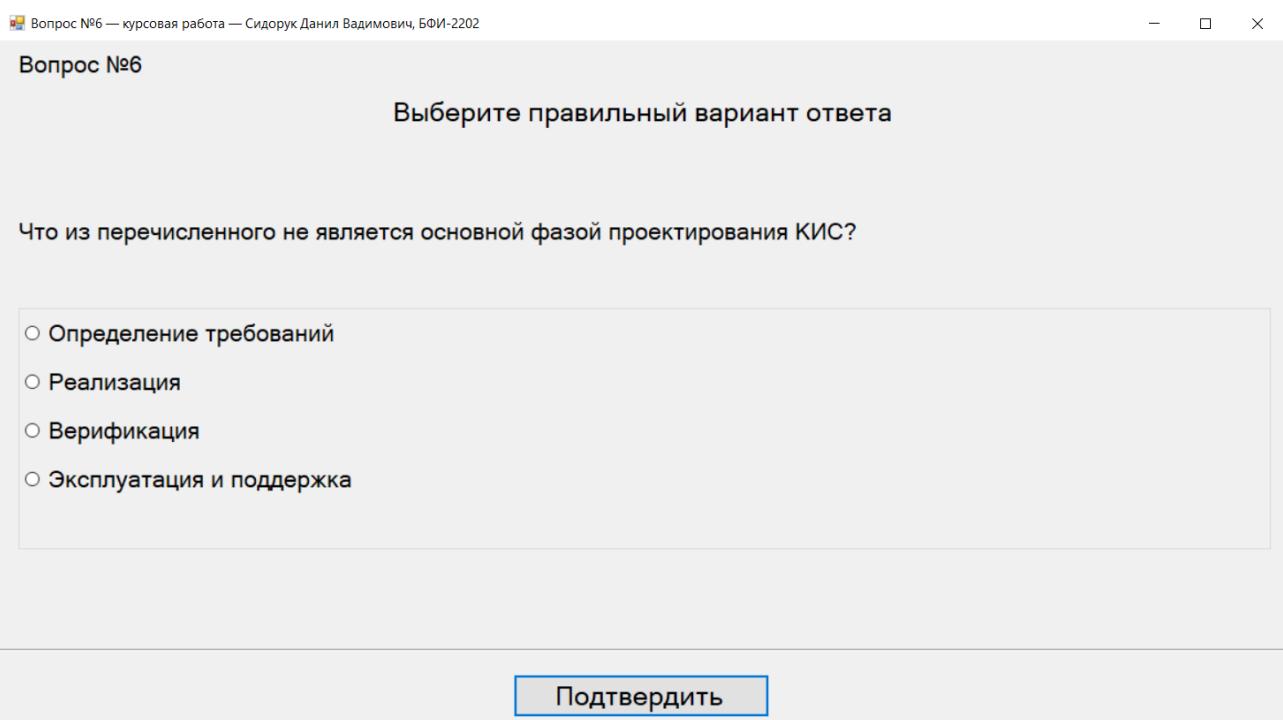


Рисунок 11 — Второй вопрос, содержащий компонент RadioButton

CheckBox – переключатель. В отличие от RadioButton этот элемент не зависит от других, что позволяет выбрать несколько вариантов ответа из представленных.

На рисунке 12 и рисунке 13 продемонстрированы формы вопросов, содержащие данный компонент.

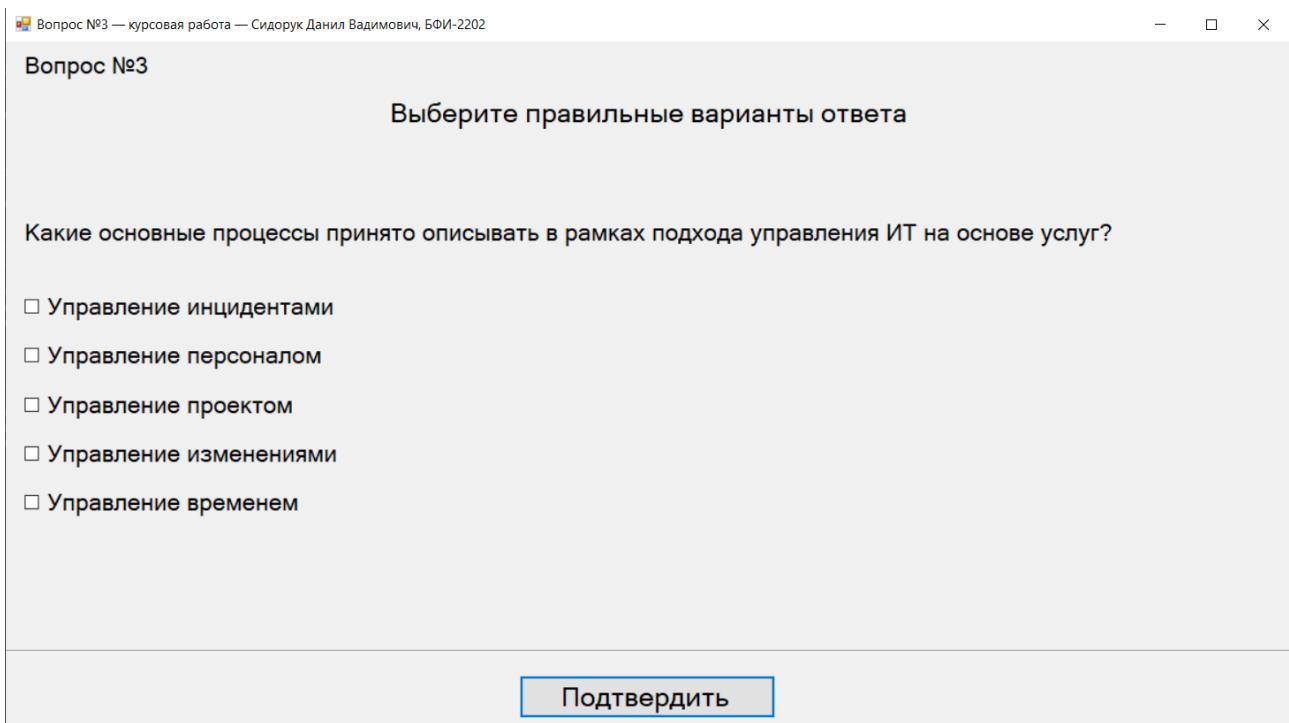


Рисунок 12 — Первый вопрос, содержащий компонент CheckBox

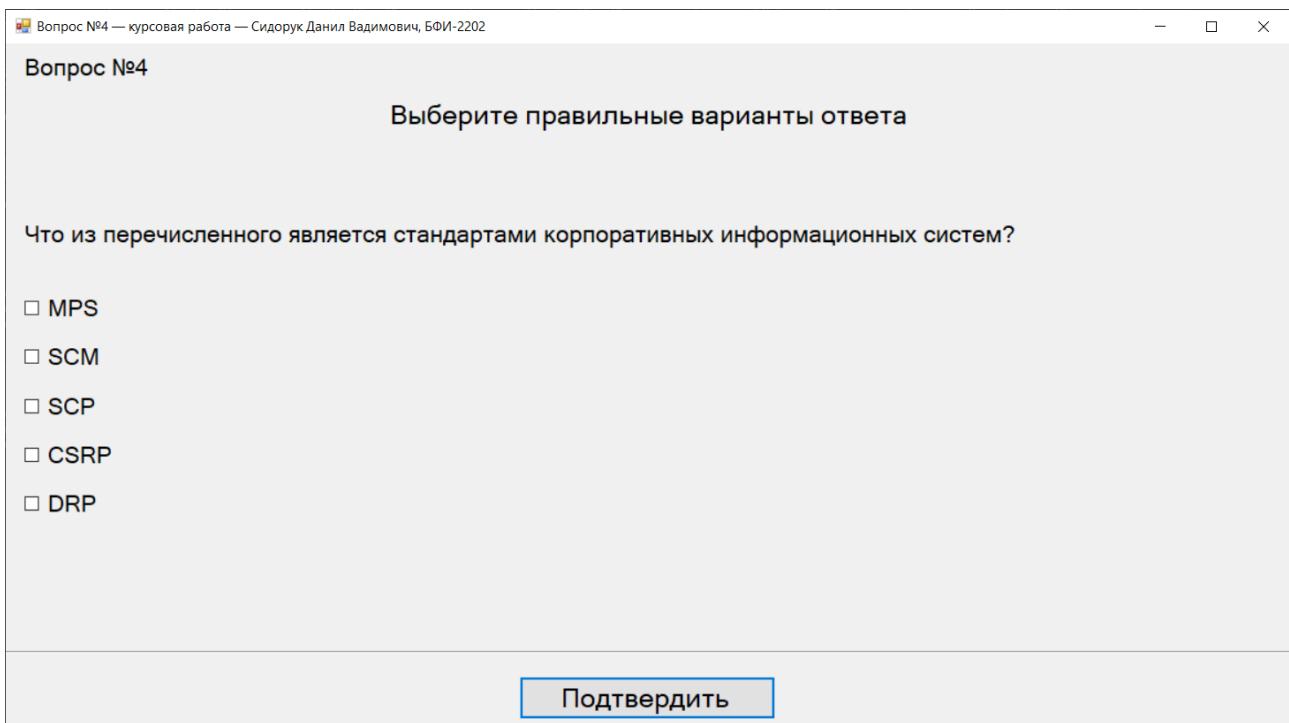


Рисунок 13 — Второй вопрос, содержащий компонент CheckBox

ListBox – список данных только для чтения. Если выбрать строки в обоих listBox, то с помощью кнопки «Объединить» можно изменить их расположение так, чтобы положение выбранного элемента во втором списке соответствовало положению выбранного элемента в первом списке. Так можно соотносить элементы одного listBox с элементами другого listBox.

На рисунке 14 и рисунке 15 продемонстрированы формы вопросов, содержащие данный компонент.

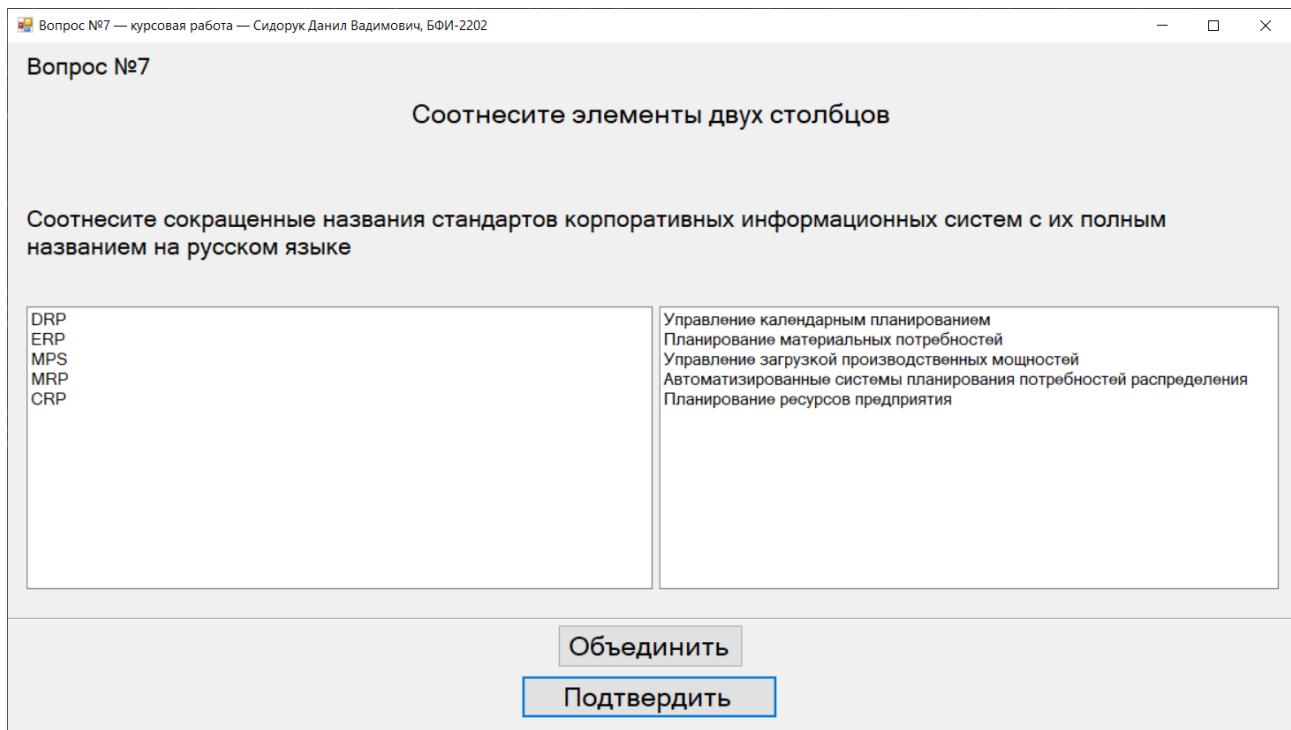


Рисунок 14 — Первый вопрос, содержащий компонент ListBox

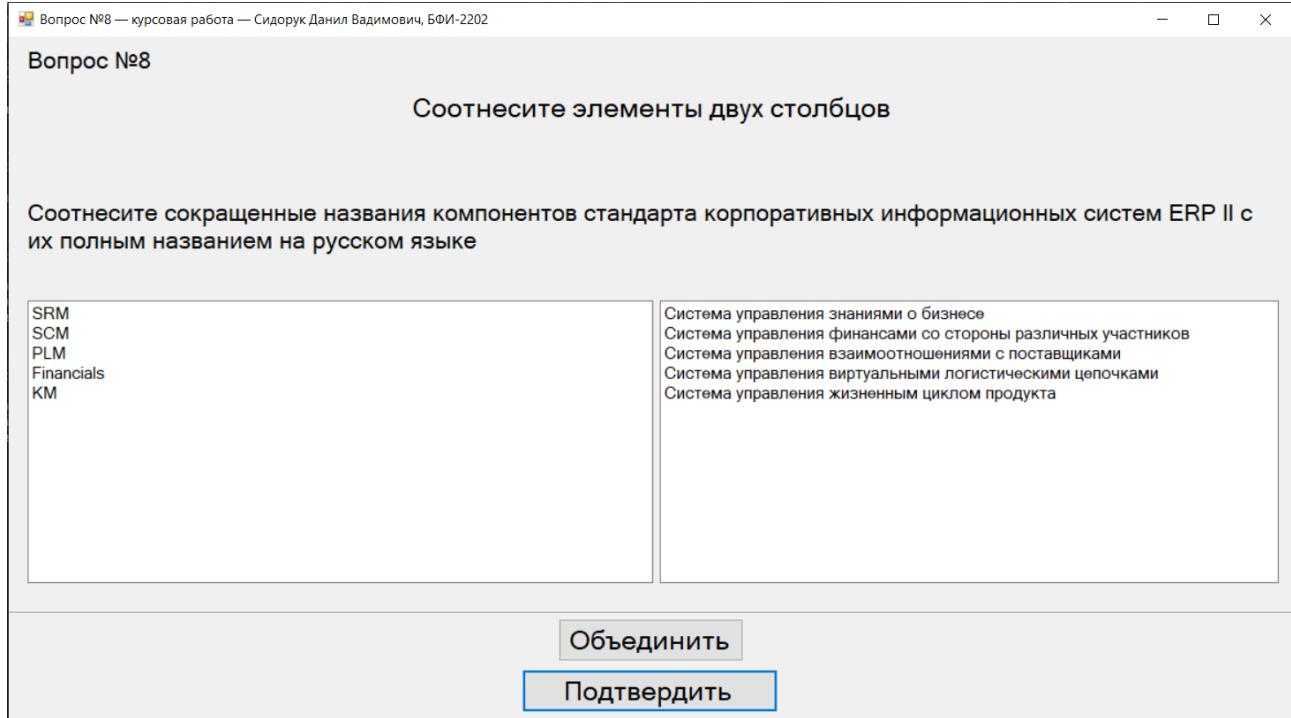


Рисунок 15 — Второй вопрос, содержащий компонент ListBox

ComboBox – выпадающий список. Как и radioButton, этот компонент позволяет выбрать один вариант ответа из нескольких предложенных. Для того, чтобы выбрать ответ, нужно нажать сначала на галочку справа, затем на любую строчку выпадающего списка.

На рисунке 16 и рисунке 17 продемонстрированы формы вопросов, содержащие данный компонент.

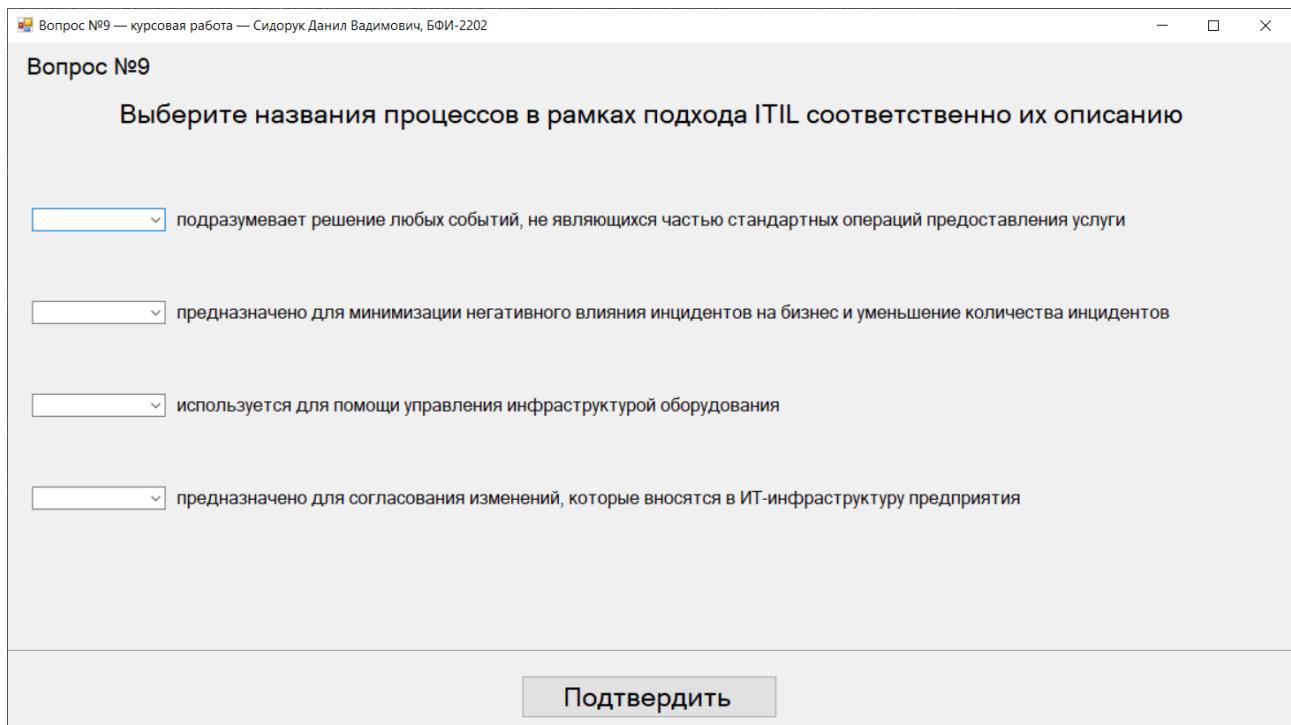


Рисунок 16 — Первый вопрос, содержащий компонент ComboBox

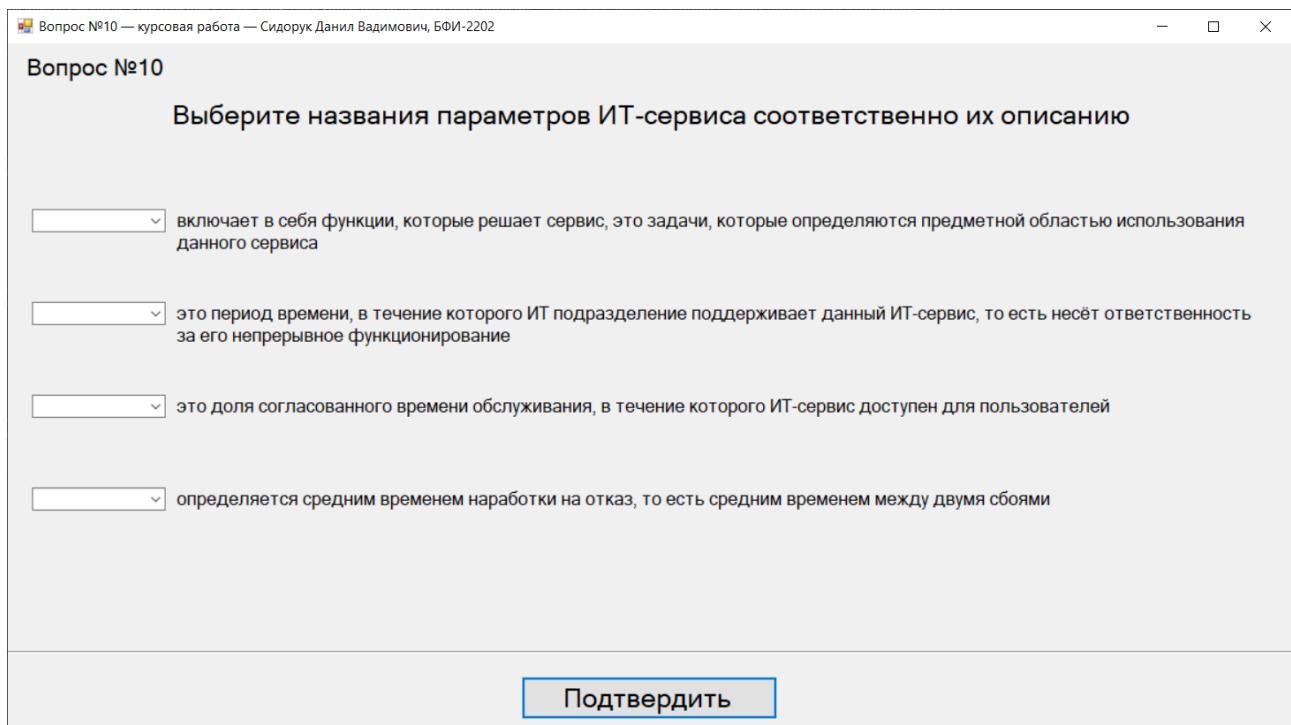


Рисунок 17 — Второй вопрос, содержащий компонент ComboBox

TrackBar – элемент управления (иногда называемый элементом управления «ползунок»), который используется для перемещения по большому объему информации или для визуальной настройки числовых параметров. Элемент управления TrackBar имеет две части: бегунок, также называемый ползунком, и метки деления.

На рисунке 18 и рисунке 19 продемонстрированы формы вопросов, содержащие данный компонент.

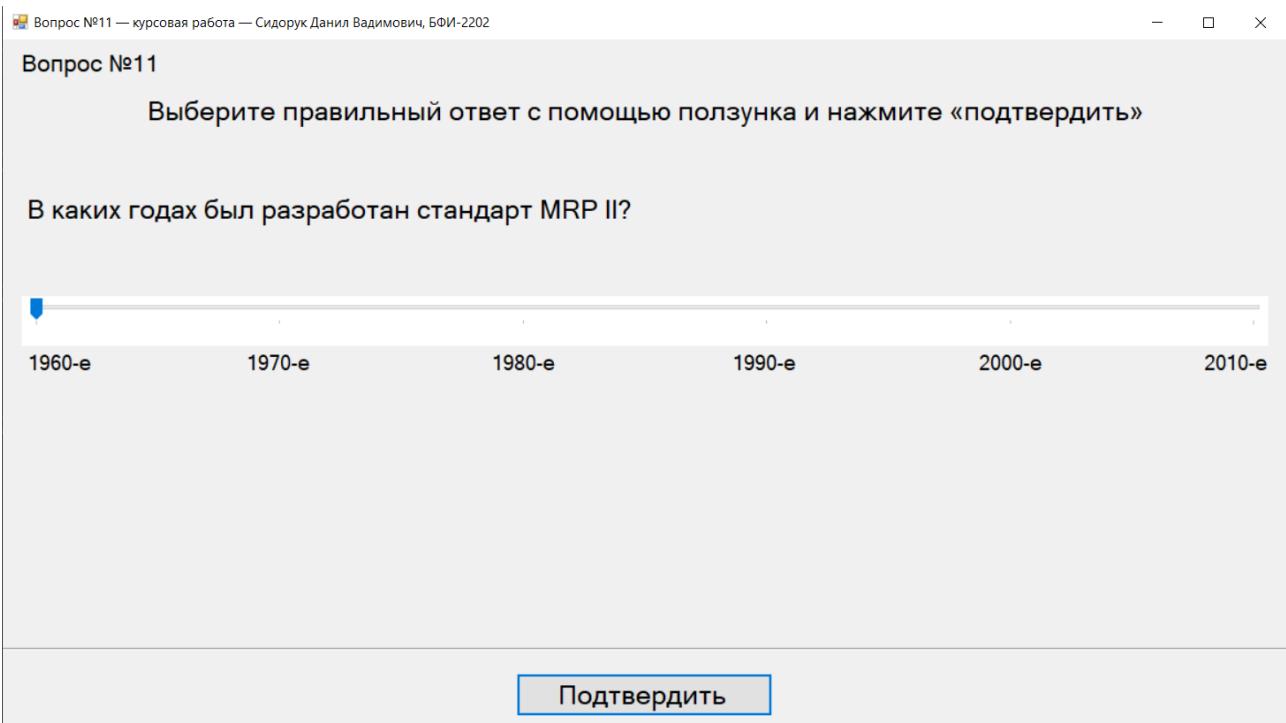


Рисунок 18 — Первый вопрос, содержащий компонент TrackBar

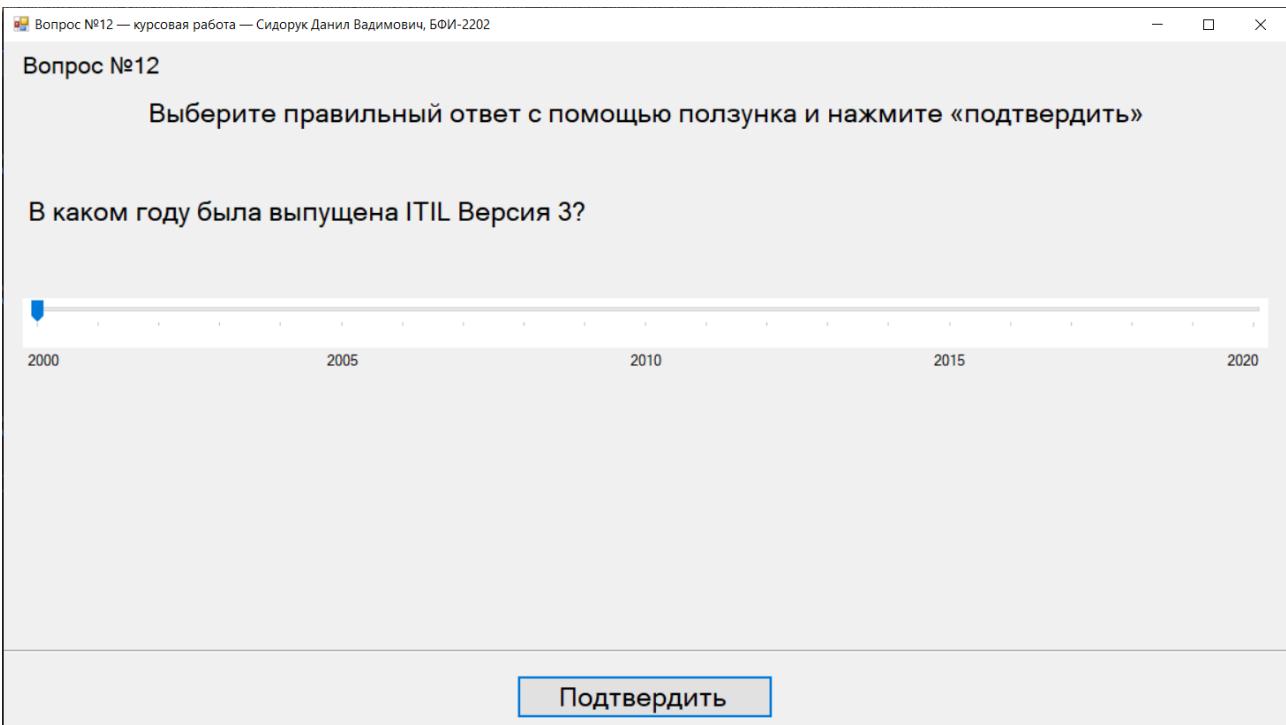


Рисунок 19 — Второй вопрос, содержащий элемент управления TrackBar

HScrollBar — элемент управления, представляющий стандартную горизонтальную полосу прокрутки Windows, использующийся, аналогично элементу управления TrackBar, для настройки числовых параметров.

На рисунке 20 и рисунке 21 продемонстрированы формы вопросов, содержащие данный компонент.

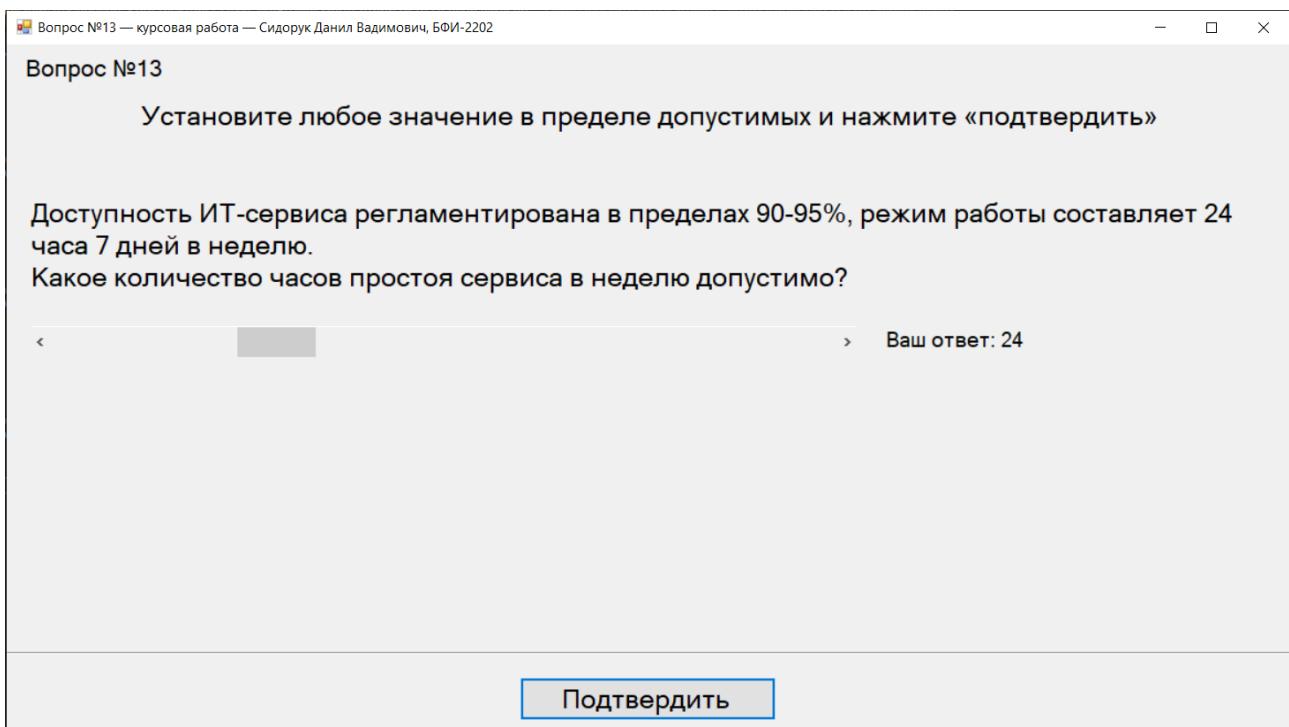


Рисунок 20 — Первый вопрос, содержащий элемент управления HScrollBar

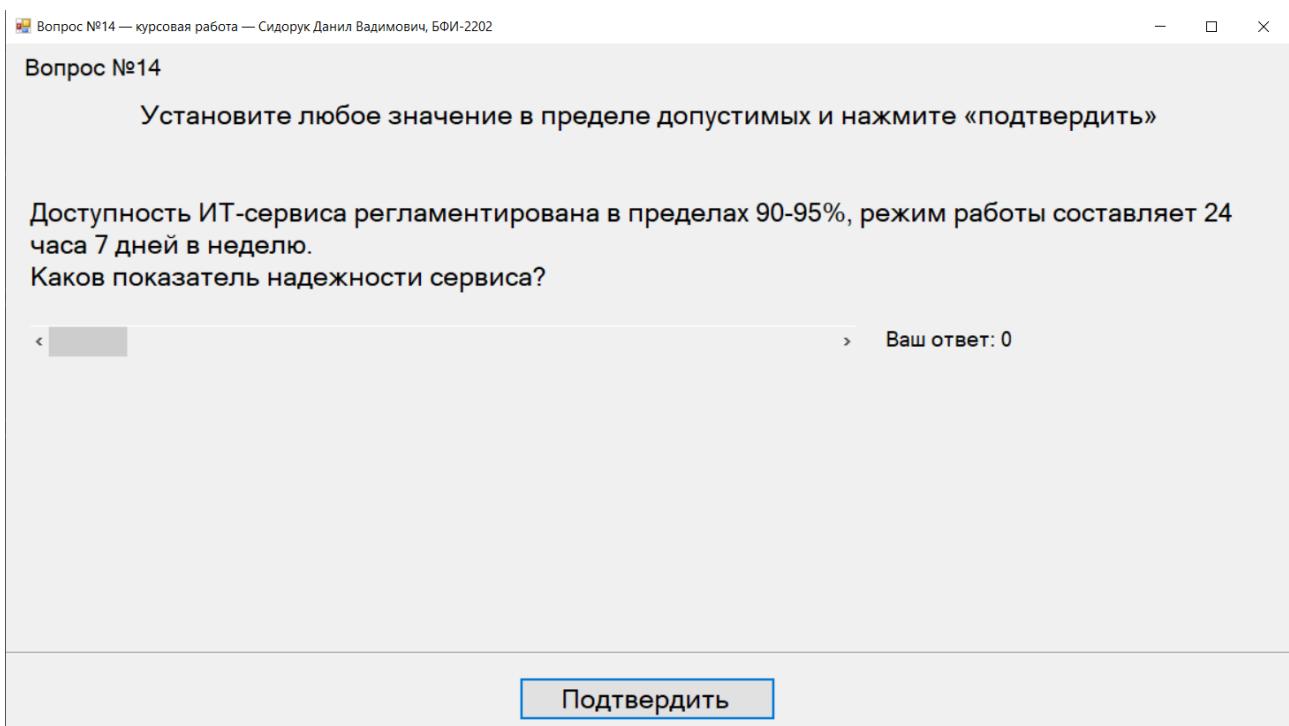


Рисунок 21 — Второй вопрос, содержащий элемент управления HScrollBar

CheckedListBox — список переключателей. Данный элемент управления позволяет выбрать несколько вариантов ответа из предложенных. Использовать CheckedListBox намного удобнее, чем несколько переключателей, поскольку так появляется возможность использовать одну и ту же форму для вопросов с разным количеством вариантов ответа.

На рисунке 22 и рисунке 23 продемонстрированы формы вопросов, содержащие данный компонент.

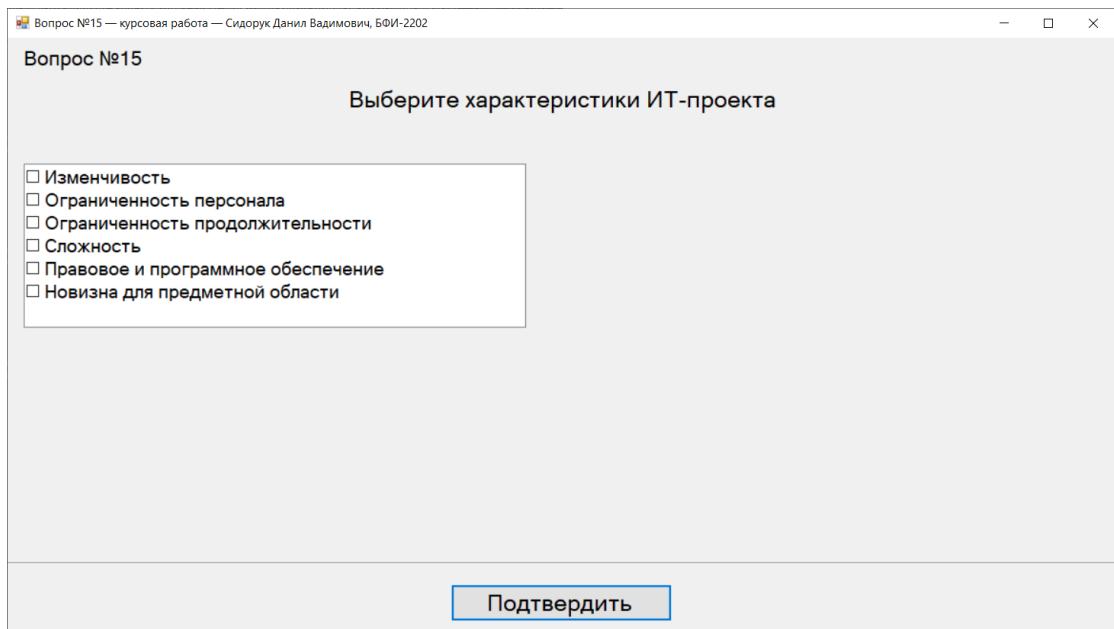


Рисунок 22 — Первый вопрос, содержащий элемент управления
CheckedListBox

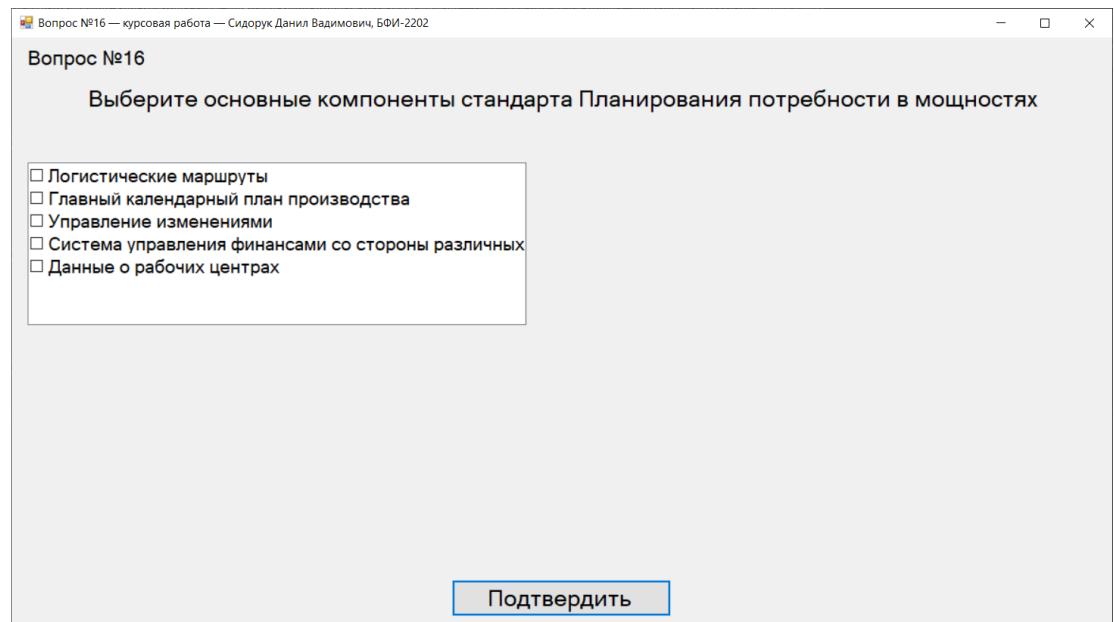
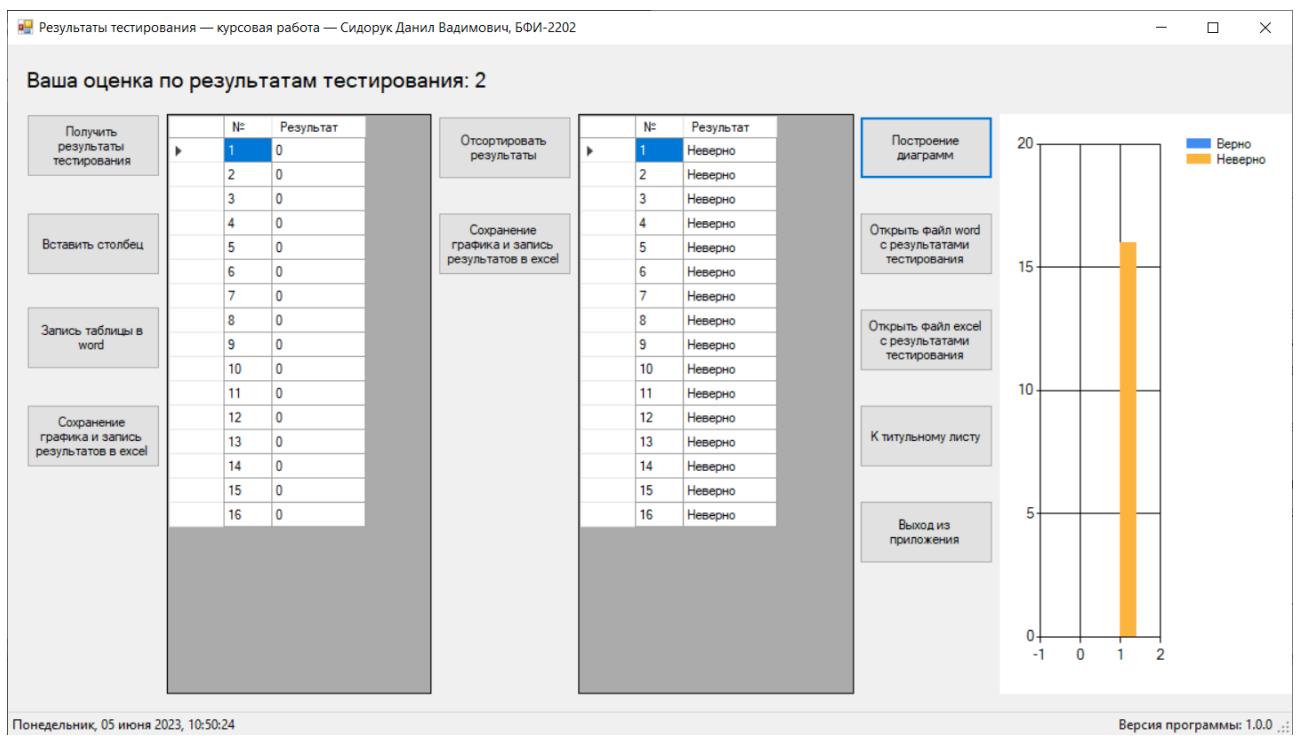


Рисунок 23 — Второй вопрос, содержащий элемент управления
CheckedListBox

После дачи ответов на все вопросы пользователя переносит на форму отображения результатов прохождения тестирования. На форме расположен Label, содержащий оценку за прохождение теста по пятибалльной шкале (менее 60% правильных ответов — два, 60–74% правильных ответов — три, 75–89% правильных ответов — четыре, 90–100% правильных ответов — пять), DataGridView, отображающий правильность ответов на конкретные вопросы, и Chart — диаграмма, с помощью которой отображено соотношение верных и неверных ответов. Также на форме расположены кнопки (button), позволяющие получить результаты тестирования, вставить новый столбец в табличный компонент с результатами тестирования, выполнить запись содержимого таблицы в документ Microsoft Word, выполнить запись содержимого данной таблицы в документ Microsoft Excel и сохранение диаграммы, отсортировать результаты тестирования с дальнейшим их выводом в табличный компонент, так же, как было описано выше, выполнить запись содержимого данной таблицы в документ Microsoft Excel и сохранение диаграммы, выполнить построение диаграммы по результатам тестирования, открыть файлы Microsoft Word и Microsoft Excel с результатами тестирования.

Также на форме расположены кнопки, позволяющие перейти на форму заставки, после чего пользователь может снова изучить теорию или пройти тест, а также выйти из приложения.

На рисунке 24 продемонстрирована форма отображения результатов прохождения тестирования.



Понедельник, 05 июня 2023, 10:50:24

Версия программы: 1.0.0 ..

Рисунок 24 — Результаты тестирования

2.3 Дополнительные возможности программы

Разработка программного проекта курсовой работы предполагала программирование дополнительных возможностей, с целью повысить оригинальность и индивидуальность работы. А именно:

- создание меню с вложенными пунктами для чтения информации из текстового файла, содержащего теоретическую часть;
- создание строки состояния с выводом текущей даты и времени, а также версии программы;
- создание медиаплеера, поддерживающий просмотр (воспроизведение) фото-, аудио- и видео- файлов;
- получение результатов тестирования в двумерный массив, его вывод в компонент DataGridView, осуществление вставки столбца из значения, заданного пользователем, запись результирующей матрицы документ Microsoft Word;
- упорядочивание результатов тестирования, представленных в виде одномерного массива в другом компоненте DataGridView, с использованием метода сортировки простым обменом (1 способ) при помощи алгоритма сортировки, показанного на занятиях. Осуществление записи результатов тестирования в документ Microsoft Excel и выполнение построения графика (диаграммы) на рабочем листе. Выполнение фотографии графика с дальнейшей ее записью по указанному пользователем пути. Осуществление нахождения суммы значений полученных результатов тестирования с использованием используя встроенной функции SUM() в программном коде метода;

— осуществление вывода результатов тестирования в документ Microsoft Excel, осуществление программным кодом построения диаграммы, отображающей результаты тестирования.

Кроме того было необходимо:

- создать анимационную заставку программы;
- с целью упрощения отладки и сопровождения программы, реализовать режим разработчика, позволяющий переходить на форму с результатами тестирования без необходимости прохождения тестирования;
- создать индикатор, отображающий прогресс прохождения тестирования;
- реализовать индикатор, отображающий количество оставшихся для решения вопросов;
- обеспечить возможность вывода упорядоченных с использованием метода сортировки простым обменом (1 способ) результатов тестирования в документ Microsoft Excel и выполнения построения графика (диаграммы) на рабочем листе, выполнения фотографии графика (диаграммы) и записи его по указанному пользователем пути;
- обеспечить возможность открытия документов Microsoft Word и Microsoft Excel с результатами тестирования;
- создать установщик программы.

Перейдем к **рассмотрению вышеперечисленных возможностей**.

При запуске проекта проигрывается анимационная заставка (см. Рисунок 25). Она была нарисована в программе GIMP, преобразована в формат gif и

затем добавлена в проект Microsoft Visual Studio. Для ее представления в программном коде была создана отдельная форма.



Рисунок 25 — Анимационная заставка

Программный код для реализации этой функциональности представлен ниже:

```
// ...
namespace AlgorithmsCourseworkGUI
{
    public partial class SplashScreenForm : Form
    {
        Timer timer = new Timer();
        public SplashScreenForm()
        {
            InitializeComponent();
            timer.Tick += SwitchToTitleForm;
            timer.Interval = 5000;
        }
    }
}
```

```

        timer.Start();
    }

    public void SwitchToTitleForm(object e, EventArgs v)
    {
        timer.Stop();

        this.Hide();

        TitleForm nextForm = new TitleForm();
        nextForm.Show();
    }
}

```

После загрузки программы с помощью сочетания клавиш Alt+F5 можно включить режим разработчика, позволяющий переходить на форму с результатами тестирования без необходимости прохождения тестирования. Для этого на титульную форму была добавлена кнопка, обеспечивающая переход на форму с результатами тестирования, по умолчанию имеющая атрибут visible, заданный в false. При нажатии вышеописанного сочетания клавиш данная кнопка становится видимой.

Программный код для реализации этой функциональности представлен ниже:

```

private void TitleForm_KeyDown(object sender, KeyEventArgs e)
{
    if (e.Alt && e.KeyCode == Keys.F5)
    {
        MessageBox.Show("Режим разработчика успешно активирован");
        resultButton.Visible = true;
        e.Handled = true;
    }
}

```

}

На все формы с вопросами был добавлен индикатор, отображающий прогресс прохождения тестирования. На рисунке 26 приведен пример формы с данным индикатором.

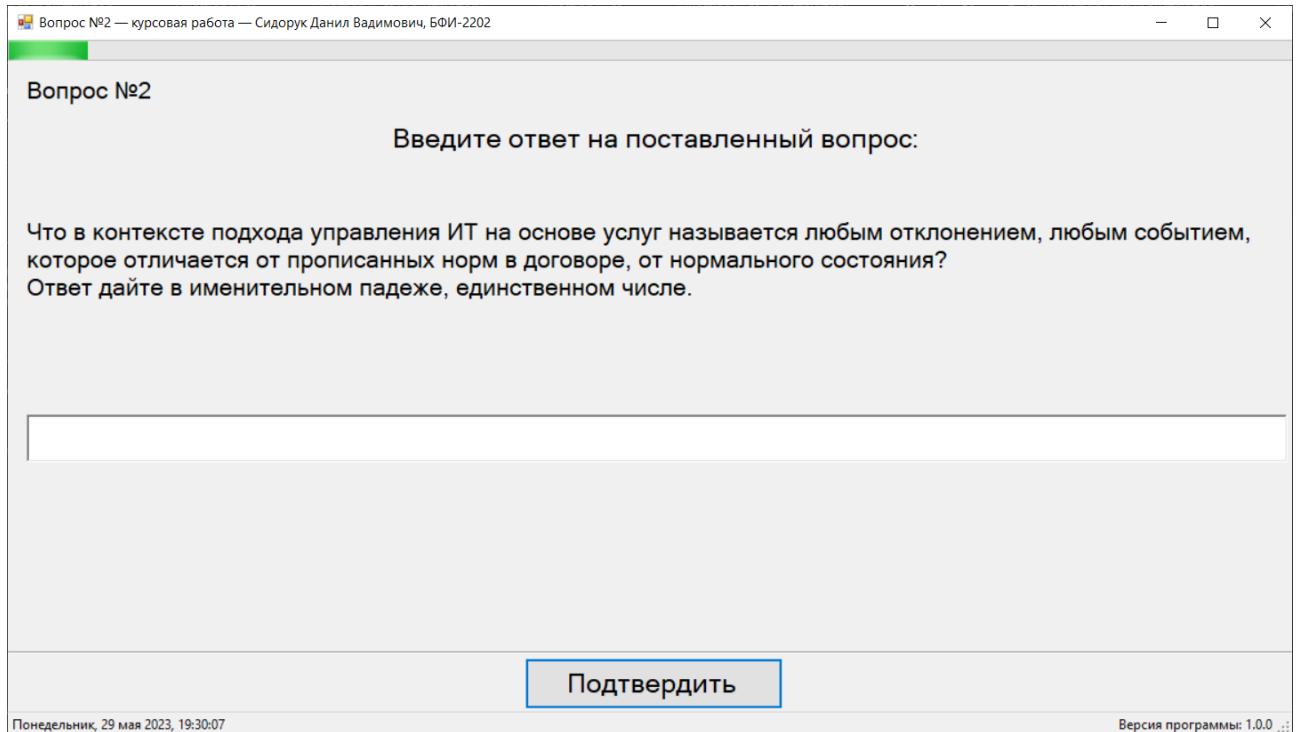


Рисунок 26 — Форма с индикатором прогресса прохождения тестирования

Для реализации данной функциональности в dll-библиотеке был реализован следующий метод:

```
public static void setupProgressBar(ProgressBar progressBar)
{
    progressBar.Minimum = 0;
    progressBar.Maximum = 16;
    progressBar.Value = n;
```

```
}
```

А в конструктор каждой формы с вопросом был добавлен следующий код:

```
AlgorithmsCoursework.setupProgressBar(progressBar);
```

На все формы с вопросами был добавлен индикатор, отображающий количество оставшихся для решения вопросов. На рисунке 27 приведен пример формы с данным индикатором.

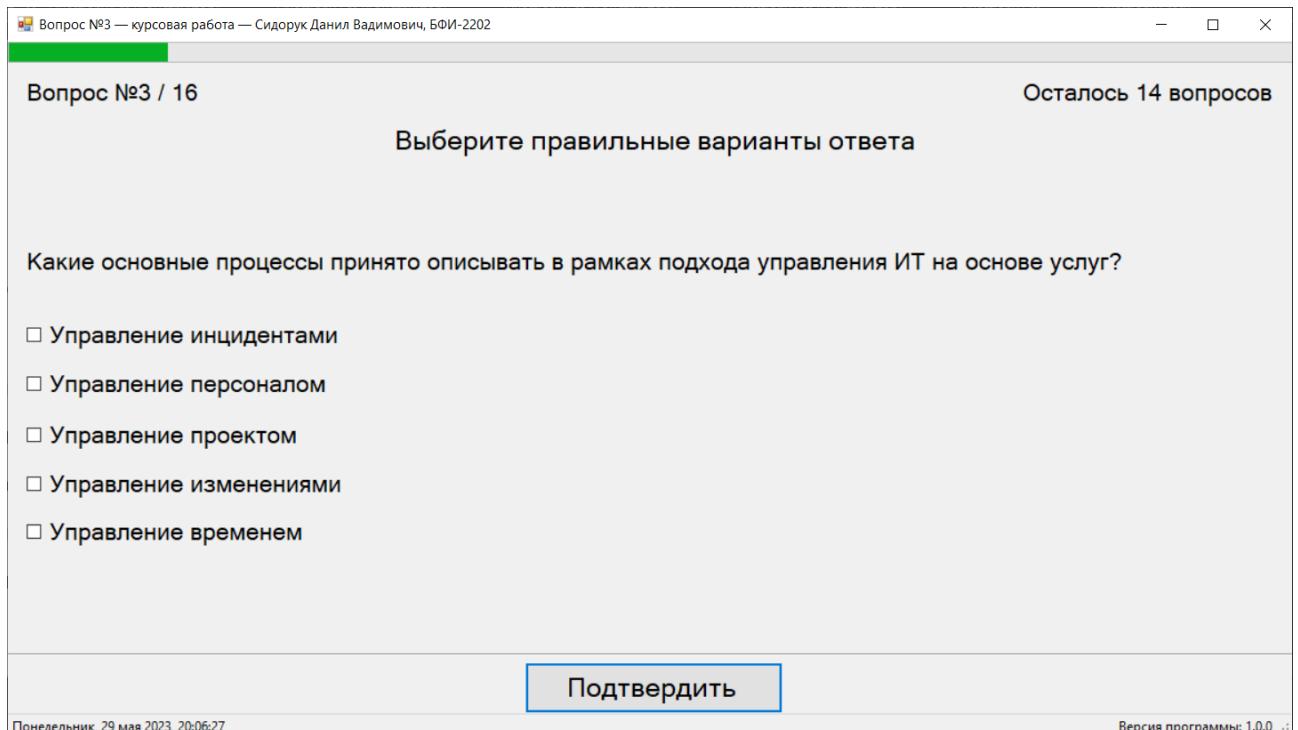


Рисунок 27 — Форма с индикатором оставшихся для решения вопросов

Для реализации данной функциональности в dll-библиотеке был реализован следующий метод:

```
public static void setupRemainingQuestionsCount(System.Windows.Forms.Label label)
```

```
{  
    label.Text = "Осталось " + (answers.Length - n) + " вопросов";  
}
```

А в конструктор каждой формы с вопросом был добавлен следующий код:

```
AlgorithmsCoursework.setupRemainingQuestionsCount(questionsCount);
```

Кроме того, на каждую форму был добавлен элемент управления StatusStrip, содержащий два ToolStripStatusLabel: один для отображения текущей даты и времени, другой — для отображения версии приложения. Текущая дата и время обновляются каждую секунду, текущая же версия приложения обновляется единожды — при инициализации формы.

На рисунке 28 приведен пример формы, содержащей данный элемент.

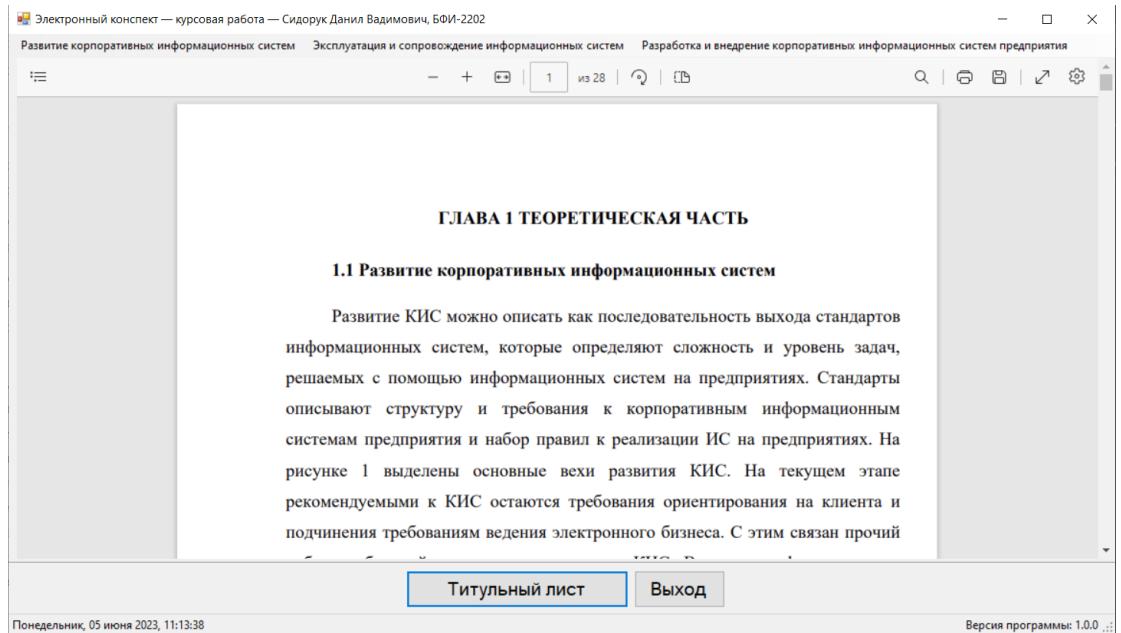


Рисунок 28 — Пример формы с строкой состояния, отображающей текущую дату, время и версию программы

Для реализации этой возможности в класс библиотеки к лабораторной работе было добавлено одно поле и реализовано три метода:

```
// AlgorithmsCoursework.cs

using System;
using System.Windows.Forms;

namespace AlgorithmsCourseworkLibrary
{
    public class AlgorithmsCoursework
    {
        // ...

        public static int version = 100;
        // ...

        public static void setupVersion(ToolStripStatusLabel label)
        {
            int version = AlgorithmsCoursework.version;
            int major = version / 100, minor = (version / 10) % 10, patch = version % 10;
            label.Text = string.Format("Версия программы: {0}.{1}.{2}", major, minor, patch);
        }

        public static void setupTimer(ToolStripStatusLabel label)
        {
            Timer timer = new Timer();
            timer.Tick += (object e, EventArgs v) => updateDate(label);
            timer.Interval = 1000;
            timer.Start();
            updateDate(label);
        }
    }
}
```

```

    }

    private static void updateDate(ToolStripStatusLabel label)
    {
        string dateInfo = DateTime.Now.ToString("ddd, dd MMMM yyyy, HH:mm:ss");
        label.Text = dateInfo.Substring(0, 1).ToUpper() + dateInfo.Substring(1);
    }
}

```

А в метод инициализации каждой формы был добавлен следующий код:

```

AlgorithmsCoursework.setupVersion(versionLabel);
AlgorithmsCoursework.setupTimer(timeLabel);

```

Кроме того, с помощью элемента управления ToolStrip, было реализовано меню для навигации по содержанию электронного конспекта.

На рисунке 29 приведен пример формы с данным меню.

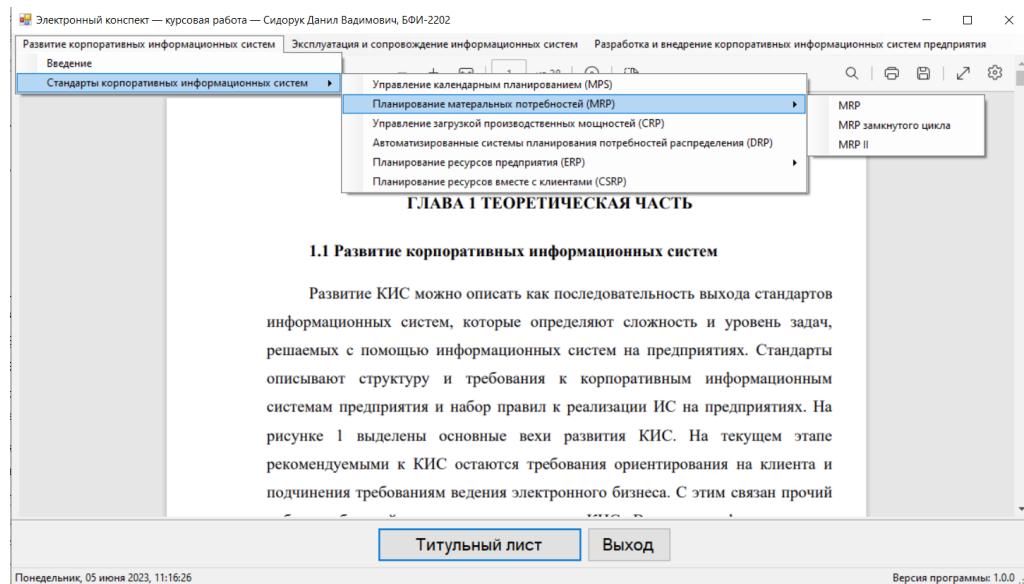


Рисунок 29 — Меню навигации по содержанию электронного конспекта

Для реализации этой возможности в классе формы с теоретической частью было реализовано два метода:

```
// TheoryForm.cs

using AlgorithmsCourseworkLibrary;
using Microsoft.Web.WebView2.Core;
using System;
using System.Windows.Forms;

namespace AlgorithmsCourseworkGUI
{
    public partial class TheoryForm : Form
    {
        // ...
        private void SetPage(int pageIndex)
        {
            string newUri = "http://localhost/report.pdf#page=" + pageIndex;
            webView.CoreWebView2.Navigate(newUri);
        }

        private void HandleItem_Click(object sender, EventArgs e)
        {
            String tag = (String)((ToolStripItem)sender).Tag;
            if (tag != null && tag.Length != 0) {
                SetPage(Convert.ToInt32(tag));
            }
        }
        // ...
    }
}
```

А каждому элементу меню был установлен тег, соответствующий номеру страницы, на которой находится содержание заголовка.

Также было реализовано упорядочивание результатов теста, предоставленных в виде одномерного массива в компоненте dataGridView, с использованием метода сортировки простым обменом, при этом использовался алгоритм сортировки, показанный на занятиях; запись результатов теста в Microsoft Excel и выполнение построения графика (диаграммы) на рабочем листе. Кроме формирования рабочей книги и графика (диаграммы) было реализовано выполнение фотографии графика с записью его по указанному пользователем пути; нахождение суммы значений полученных результатов тестирования с использованием встроенной функции SUM(); вывод результатов тестирования в Microsoft Excel и построение диаграммы, отображающей результаты тестирования.

На рисунке 30 демонстрируется корректность выполнения сортировки.

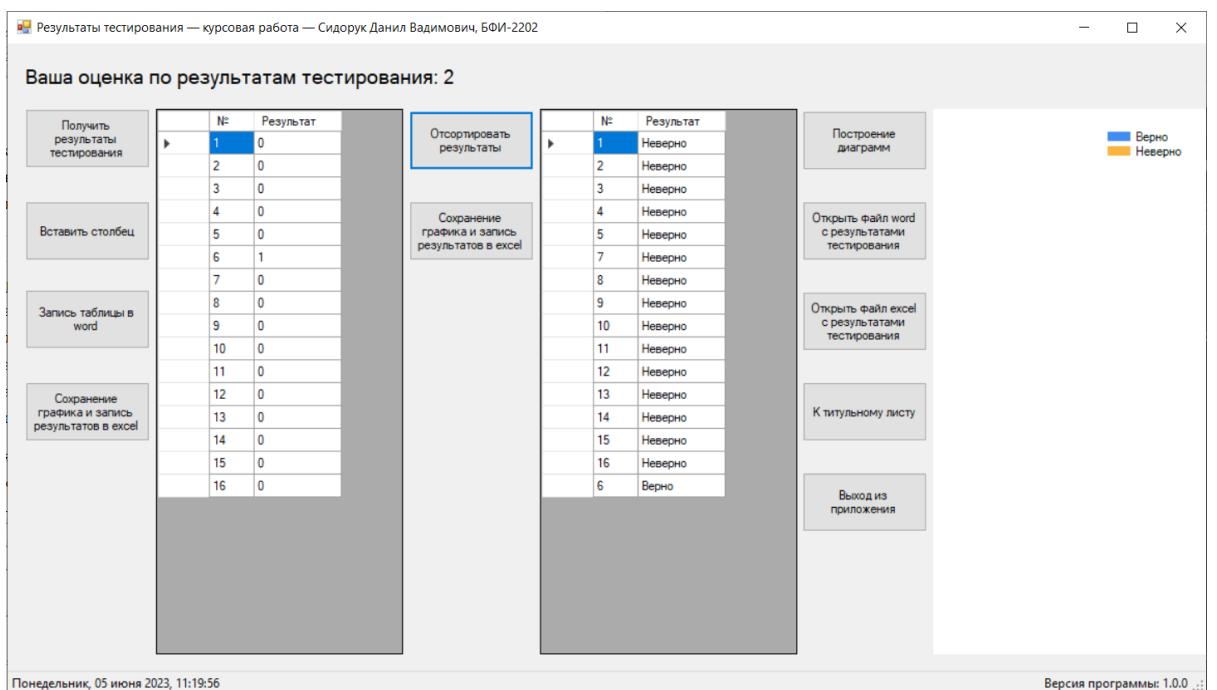


Рисунок 30 — Сортировка результатов тестирования

На рисунке 31 демонстрируется корректность вывода отсортированных результатов тестирования в документ Microsoft Excel.

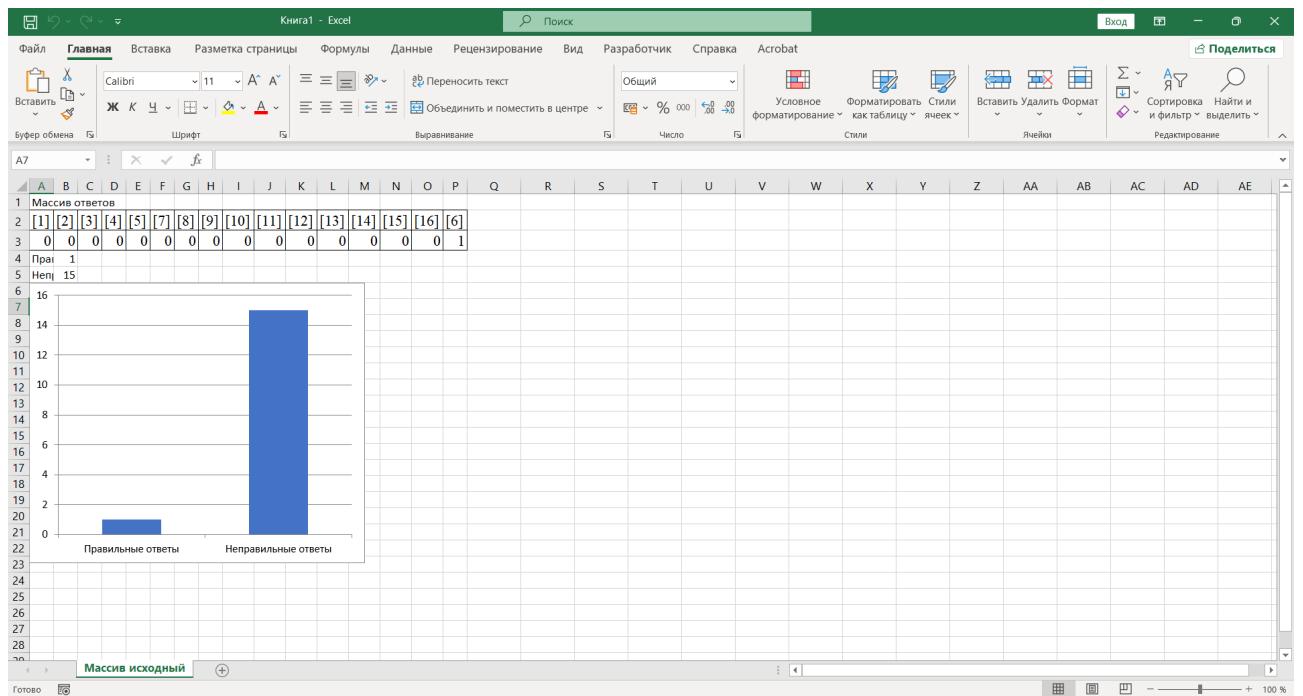


Рисунок 31 — Вывод отсортированных результатов тестирования в документ Microsoft Excel

Для выполнения данной задачи в классе формы с результатами было добавлено два поля и реализовано три метода:

```
using System;
using System.Linq;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;
```

```

namespace AlgorithmsCourseworkGUI
{
    public partial class ResultForm : Form
    {
        int[] sortedKeys = null;
        bool[] sortedValues = null;

        // ...

        private void writeExcelButton_Click(object sender, EventArgs e)
        {
            AlgorithmsCoursework.WriteExcel(
                Enumerable.Range(1, AlgorithmsCoursework.answers.Length).ToArray(),
                AlgorithmsCoursework.answers
            );
        }

        private void sortButton_Click(object sender, EventArgs e)
        {
            sortedKeys = new int[AlgorithmsCoursework.answers.Length];
            sortedValues = new bool[AlgorithmsCoursework.answers.Length];
            for (int i = 0; i != AlgorithmsCoursework.answers.Length; ++i)
            {
                sortedKeys[i] = i + 1;
                sortedValues[i] = AlgorithmsCoursework.answers[i];
            }

            AlgorithmsCoursework.Sort(sortedKeys, sortedValues);
            AlgorithmsCoursework.OutputRows(sortedView, sortedKeys, sortedValues);
        }

        private void writeSortExcelButton_Click(object sender, EventArgs e)
    }
}

```

```

{
    if (sortedKeys == null || sortedValues == null)
    {
        MessageBox.Show("Выполните сортировку", "Операция невозможна",
        MessageBoxButtons.OK, MessageBoxIcon.Error);

        return;
    }

    AlgorithmsCoursework.Sort(sortedKeys, sortedValues);
    AlgorithmsCoursework.WriteExcel(sortedKeys, sortedValues);
}

// ...
}

```

И четыре метода в dll-библиотеке:

```

using System;
using System.Drawing;
using System.Windows.Forms;

using Microsoft.Office.Interop.Excel;
using Microsoft.Office.Interop.Word;
using CheckBox = System.Windows.Forms.CheckBox;

namespace AlgorithmsCourseworkLibrary
{
    public class AlgorithmsCoursework
    {
        // ...

        public static void OutputRows(DataGridView view, int[] keys, bool[] values)
        {
            view.Rows.Clear();
        }
    }
}

```

```

        for (int i = 0; i != keys.Length; i++)
    {
        view.Rows.Add(keys[i], values[i] == true ? "Верно" : "Неверно");
    }
}

public static void SaveChartAsImage(ChartObject chart)
{
    chart.CopyPicture(
        Microsoft.Office.Interop.Excel.XlPictureAppearance.xlScreen,
        Microsoft.Office.Interop.Excel.XlCopyPictureFormat.xlBitmap
    );
    Bitmap image = new Bitmap(Clipboard.GetImage());

    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.Filter = "Image (*.jpg) | *.jpg";
    saveFileDialog.Title = "Сохранить изображение";

    while (saveFileDialog.ShowDialog() != DialogResult.OK)
        continue;

    image.Save(saveFileDialog.FileName,
    System.Drawing.Imaging.ImageFormat.Jpeg);
}

public static void WriteExcel(int[] keys, bool[] values)
{
    Microsoft.Office.Interop.Excel.Application app = new
Microsoft.Office.Interop.Excel.Application();
    Workbook workBook;
    Worksheet workSheet;
}

```

```

workBook = app.Workbooks.Add();

workSheet = (Worksheet)workBook.Worksheets.get_Item(1);

workSheet.Name = "Массив исходный";
workSheet.Cells[1, 1] = "Массив ответов";
for (int i = 0; i != values.Length; i++)
{
    workSheet.Cells[2, i + 1] = "[" + keys[i] + "]";
    workSheet.Cells[3, i + 1] = Convert.ToInt32(values[i]);
}

Microsoft.Office.Interop.Excel.Range range =
workSheet.Range[workSheet.Cells[2, 1], workSheet.Cells[3, values.Length]];
range.Cells.Font.Name = "Times New Roman";
range.Cells.Font.Size = 14;
range.Cells.Columns.AutoFit();
range.Borders.get_Item(XlBordersIndex.xlEdgeBottom).LineStyle =
    Microsoft.Office.Interop.Excel.XlLineStyle.xlContinuous;
range.Borders.get_Item(XlBordersIndex.xlEdgeRight).LineStyle =
    Microsoft.Office.Interop.Excel.XlLineStyle.xlContinuous;
range.Borders.get_Item(XlBordersIndex.xlInsideHorizontal).LineStyle =
    Microsoft.Office.Interop.Excel.XlLineStyle.xlContinuous;
range.Borders.get_Item(XlBordersIndex.xlInsideVertical).LineStyle =
    Microsoft.Office.Interop.Excel.XlLineStyle.xlContinuous;
range.Borders.get_Item(XlBordersIndex.xlEdgeTop).LineStyle =
    Microsoft.Office.Interop.Excel.XlLineStyle.xlContinuous;

workSheet.Cells[4, 1] = "Правильные ответы";
workSheet.Cells[4, 2] = "=SUM(A3:P3)";
workSheet.Cells[5, 1] = "Неправильные ответы";
workSheet.Cells[5, 2] = "=" + values.Length + "-SUM(A3:P3)";

object misValue = System.Reflection.Missing.Value;

```

```

Microsoft.Office.Interop.Excel.Range chartRange;

ChartObjects xlCharts =
(ChartObjects)workSheet.ChartObjects(Type.Missing);

ChartObject myChart = (ChartObject)xlCharts.Add(0, 80, 300, 250);

Microsoft.Office.Interop.Excel.Chart chartPage = myChart.Chart;

chartRange = workSheet.get_Range("A4", "B5");

chartPage.SetSourceData(chartRange, misValue);

chartPage.ChartType = XlChartType.xlColumnClustered;

chartPage.Legend.Delete();

SaveChartAsImage(myChart);

workSheet.Range[("A7")].Select();

app.Visible = true;

app.UserControl = true;

}

public static void Sort(int[] keys, bool[] values)
{
    for (int j = 1; j != values.Length; ++j)
    {
        var key = keys[j];
        var value = values[j];
        var i = j - 1;

        while (i >= 0 && Convert.ToInt32(values[i]) > Convert.ToInt32(value))
        {
            keys[i + 1] = keys[i];
            values[i + 1] = values[i];
        }
    }
}

```

```

        i = i - 1;

    }

    keys[i + 1] = key;
    values[i + 1] = value;
}

}

// ...
}

```

Было реализовано получение результатов записи ответов на тест в двумерный массив с выводом в компонент dataGridView, осуществление вставки столбца из значения, заданного пользователем; запись результирующей матрицы в редактор Microsoft Word.

На рисунке 32 продемонстрирован результат вставки нового столбца с названием «Столбец» (название было задано пользователем).

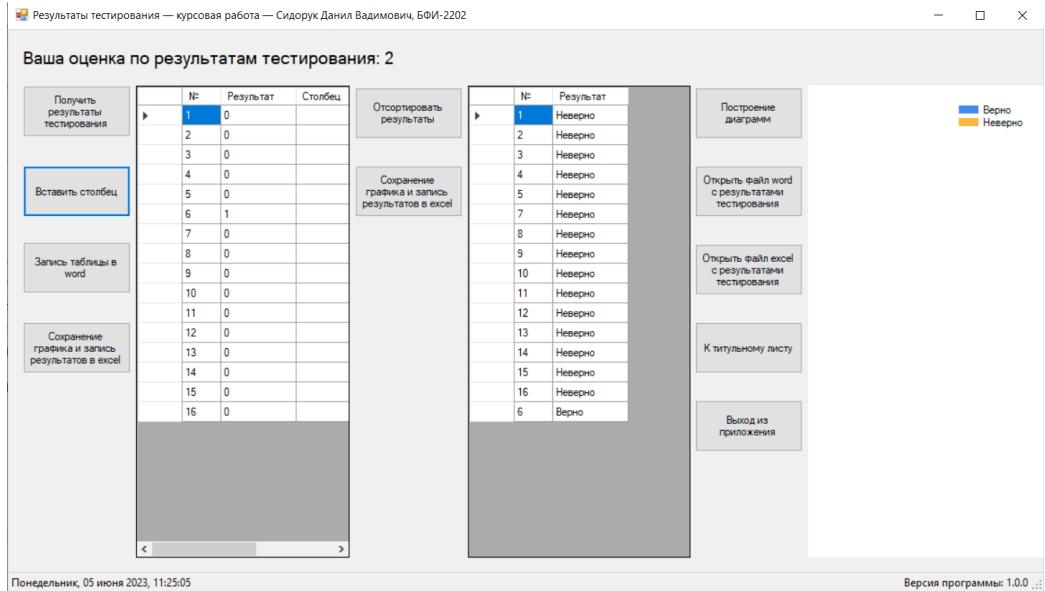


Рисунок 32 — Результат вставки нового столбца

На рисунке 33 демонстрируется корректность записи результирующей матрицы в документ Microsoft Word.

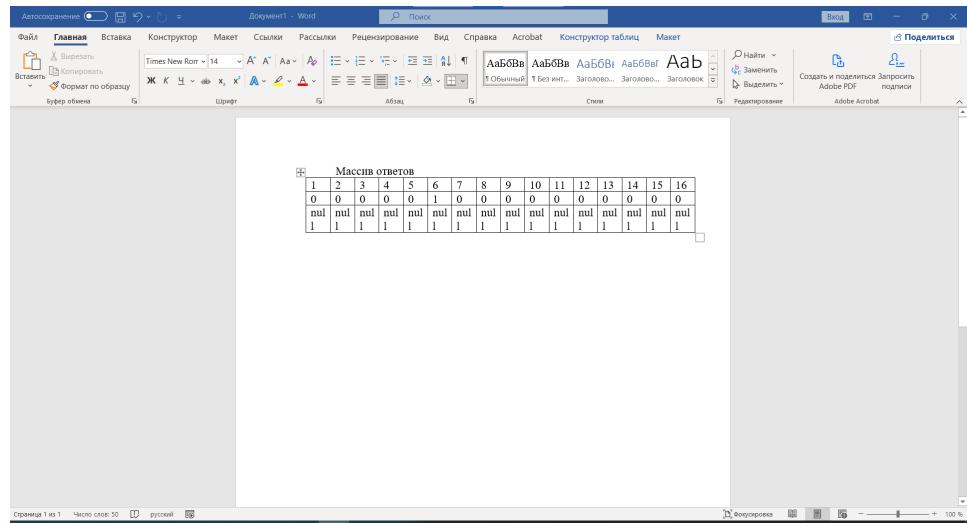


Рисунок 33 — Результат записи результирующей матрицы в документ Microsoft Word

Для выполнения данной задачи было реализовано три метода в классе формы с результатом:

```
using System;
using System.Linq;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class ResultForm : Form
```

```

{

    // ...




    private void fetchButton_Click(object sender, EventArgs e)
    {
        int[,] resultsMatrix = AlgorithmsCoursework.GetResultsMatrix();
        AlgorithmsCoursework.OutputRows(resultView, resultsMatrix);
    }






    private void writeButton_Click(object sender, EventArgs e)
    {
        if (resultView.RowCount == 0)
        {
            MessageBox.Show("Получите результаты тестирования", "Операция невозможна", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }

        AlgorithmsCoursework.WriteWord(resultView);
    }






    private void insertColumnButton_Click(object sender, EventArgs e)
    {
        string name = "";
        while (name == "")
            name = Microsoft.VisualBasic.Interaction.InputBox("Пожалуйста, введите название нового столбца", "Создание столбца");




        var column = new DataGridViewTextBoxColumn();
        column.ReadOnly = false;
        column.Name = name;
        resultView.Columns.Add(column);
    }
}

```

```
    }

}

}
```

А также три метода в dll-библиотеке:

```
using System;
using System.Drawing;
using System.Windows.Forms;

using Microsoft.Office.Interop.Excel;
using Microsoft.Office.Interop.Word;
using CheckBox = System.Windows.Forms.CheckBox;

namespace AlgorithmsCourseworkLibrary
{
    public class AlgorithmsCoursework
    {
        // ...

        public static int[,] GetResultsMatrix()
        {
            int[,] resultsMatrix = new int[answers.Length, 2];
            for (int i = 0; i != answers.Length; i++)
            {
                resultsMatrix[i, 0] = i + 1;
                resultsMatrix[i, 1] = answers[i] == true ? 1 : 0;
            }
            return resultsMatrix;
        }

        public static void OutputRows(DataGridView view, int[,] resultsMatrix)
        {
            view.Rows.Clear();
```

```

        for (int i = 0; i != resultsMatrix.GetLength(0); ++i)
        {
            view.Rows.Add(resultsMatrix[i, 0], resultsMatrix[i, 1]);
        }
    }

    public static void WriteWord(DataGridView view)
    {
        Microsoft.Office.Interop.Word.Application app = new
Microsoft.Office.Interop.Word.Application();
        var Word = new Microsoft.Office.Interop.Word.Application
        {
            Visible = true,
        };
        var inf = Type.Missing;
        var Document = Word.Documents.Add(inf);
        Word.Selection.TypeText("Массив ответов");

        object t1 = WdDefaultTableBehavior.wdWord9TableBehavior;
        object t2 = WdAutoFitBehavior.wdAutoFitContent;

        int length = view.RowCount, columns = view.ColumnCount;
        Table tbl = Word.ActiveDocument.Tables.Add(Word.Selection.Range, columns,
length, t1, t2);

        for (int i = 0; i != columns; ++i)
        {
            for (int j = 0; j != length; ++j)
            {
                object value = view.Rows[j].Cells[i].Value;
                tbl.Cell(i + 1, j + 1).Range.InsertAfter(value == null ? "null":
value.ToString());
            }
        }
    }
}

```

```
        }

    }

    // ...

}

}
```

Также на отдельной форме был реализован медиаплеер.

На рисунке 34 продемонстрирован интерфейс данной формы:

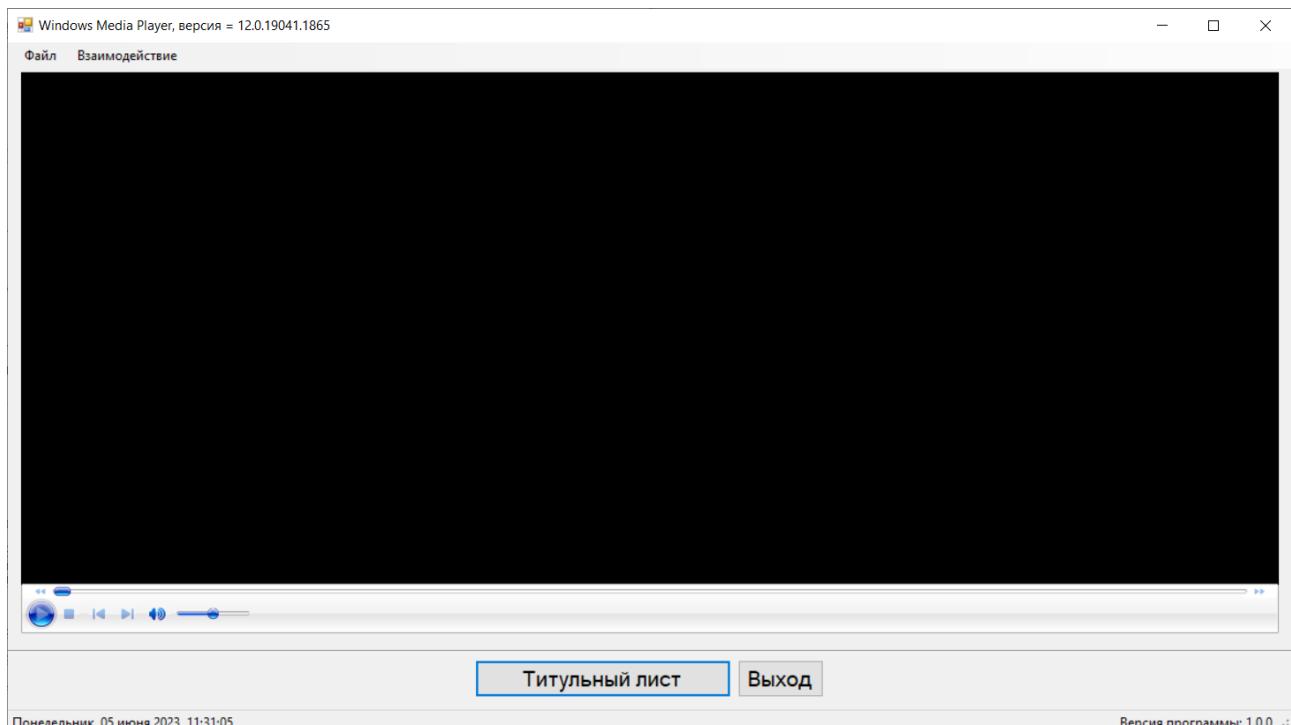


Рисунок 34 — Медиаплеер

Ниже приведен код формы, содержащей медиаплеер:

```
using AlgorithmsCourseworkLibrary;
using System;
using System.Windows.Forms;

namespace AlgorithmsCourseworkGUI
{
    public partial class PlayerForm : Form
    {
```

```

 OpenFileDialog openFileDialog;

public PlayerForm()
{
    InitializeComponent();

    AlgorithmsCoursework.setupVersion(versionLabel);
    AlgorithmsCoursework.setupTimer(timeLabel);
}

private void PlayerForm_Load(object sender, EventArgs e)
{
    openFileDialog = new OpenFileDialog();
    this.Text = "Windows Media Player, версия = " + player.versionInfo;
}

private void exitButton_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void titleButton_Click(object sender, EventArgs e)
{
    this.Hide();
    TitleForm nextForm = new TitleForm();
    nextForm.Show();
}

private void открытьToolStripMenuItem_Click(object sender, EventArgs e)
{
    openFileDialog.ShowDialog();
    player.URL = openFileDialog.FileName;
    player.Ctlcontrols.play();
}

private void выходToolStripMenuItem_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void полноэкранныйРежимToolStripMenuItem_Click(object sender,
EventArgs e)
{
    if (player.playState == WMPLib.WMPPlayState.wmppsPlaying)
        player.fullScreen = true;
}

private void паузаToolStripMenuItem_Click(object sender, EventArgs e)
{
    player.Ctlcontrols.pause();
}

private void начатьВоспроизведениеToolStripMenuItem_Click(object sender,
EventArgs e)
{
    player.Ctlcontrols.play();
}

```

```

    private void выключитьвключитьЗвукToolStripMenuItem_Click(object sender,
EventArgs e)
{
    player.settings.mute = !player.settings.mute;
}

private void свойстваToolStripMenuItem_Click(object sender, EventArgs e)
{
    player.ShowPropertyPages();
}
}
}

```

На форму с результатами тестирования были добавлены кнопки, предоставляющие возможность открытия документов Microsoft Word и Microsoft Excel с результатами тестирования.

Для реализации этой возможности в классе формы с результатами тестирования были созданы два метода:

```

private void openButton_Click(object sender, EventArgs e)
{
    AlgorithmsCoursework.OpenFile("Microsoft Word (*.docx)|*.docx|All files (*.*)|*.*",
        "Выберите документ Word");
}

private void openExcelButton_Click(object sender, EventArgs e)
{
    AlgorithmsCoursework.OpenFile(
        "Excel Workbook (*.xlsx)|*.xlsx|Excel Macro-Enabled Workbook (*.xlsm)|*.xlsm|All files (*.*)|*.*",
        "Выберите документ Excel"
    );
}

```

А также один метод в dll-библиотеке:

```

public static void OpenFile(string filters, string title)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();

```

```
openFileDialog.Filter = filters;
openFileDialog.Title = title;
if (openFileDialog.ShowDialog() != DialogResult.OK)
{
    MessageBox.Show("Вы не выбрали файл", "Открыть", MessageBoxButtons.OK,
MessageBoxIcon.Error);
    return;
}
System.Diagnostics.Process.Start(openFileDialog.FileName);
}
```

Кроме того, был создан установщик. Данная задача была выполнена с помощью стандартного расширения «Microsoft Visual Studio Installer Projects». Был создан новый проект по шаблону «Setup Wizard». Были сконфигурированы файлы, копирующиеся в директорию программы. Были настроены ярлыки, создающиеся на рабочем столе и меню программ пользователя по установке программы.

На рисунке 35 продемонстрирована работа установщика:

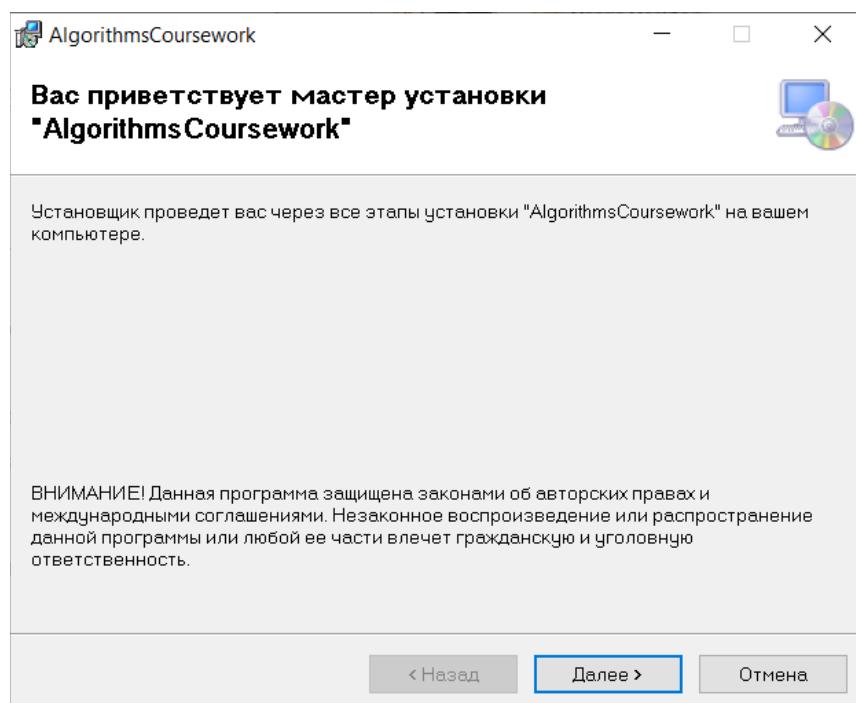


Рисунок 35 — Работа установщика

По итогам работы над второй главой можно отметить, что был разработан электронный конспект и создана программа для тестирования пользователя по теме «Эксплуатация корпоративных информационных систем». При этом были выполнены все поставленные задачи.

ЗАКЛЮЧЕНИЕ

В рамках работы над курсовым проектом был разработан программный продукт, который предназначен для предоставления возможности дистанционного изучения материала по теме курсовой работы, а также для проверки знаний учащихся с помощью тестовой программы.

В результате выполненной работы были решены следующие задачи:

- проанализирован и структурирован теоретический материал по теме исследования;
- разработан электронный конспект и программа, проверяющая усвоение знаний по данной теме;
- создано меню с вложенными пунктами для чтения информации из текстового файла, содержащего теоретическую часть;
- создана строка состояния с выводом текущей даты и времени, а также версии программы;
- создан медиаплеер, поддерживающий просмотр (воспроизведение) фото-, аудио- и видео- файлов;
- реализована запись результатов тестирования в двумерный массив, вывод его в компонент DataGridView, осуществление вставки столбца из значения, заданного пользователем, вывод результирующей матрицы в документ Microsoft Word;
- реализовано упорядочивание результатов тестирования, представленных в виде одномерного массива в другом компоненте DataGridView, с использованием метода сортировки простым обменом (1

способ), осуществление записи результатов тестирования в Microsoft Excel и выполнение построения графика (диаграммы) на рабочем листе, выполнение фотографии графика с последующей ее записью по указанному пользователем пути, осуществление нахождения суммы значений полученных результатов тестирования с использованием встроенной функции SUM() в программном коде метода;

– реализовано осуществление вывода результатов тестирования в Microsoft Excel, построение диаграммы, отображающей результаты тестирования.

– реализован, с целью упрощения отладки и сопровождения программы, режим разработчика, позволяющий переходить на форму с результатами тестирования без необходимости прохождения тестирования;

– создан индикатор, отображающий прогресс прохождения тестирования;

– реализован индикатор, отображающий количество оставшихся для решения вопросов;

– обеспечена возможность вывода упорядоченных с использованием метода сортировки простым обменом (1 способ) результатов тестирования в документ Microsoft Excel и выполнения построения графика (диаграммы) на рабочем листе, выполнения фотографии графика (диаграммы) и записи его по указанному пользователем пути;

– обеспечена возможность открытия документов Microsoft Word и Microsoft Excel с результатами тестирования;

– создан установщик программы.

Таким образом, цели и задачи, поставленные на курсовую работу, были успешно выполнены.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Гантц, И. С. Эксплуатация корпоративных информационных систем : учебное пособие / И. С. Гантц. — Москва : РТУ МИРЭА, 2022. — 75 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/256730> (дата обращения: 15.05.2023). — Режим доступа: для авториз. Пользователей.
2. Гуриков С. Р. Введение в программирование на языке Visual C# : учебное пособие / С. Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2020. — 447 с.
3. Гуриков, С. Р. Введение в программирование на языке Visual Basic for Applications (VBA) : учебное пособие / С.Р. Гуриков. - Москва : ИНФРА-М, 2021. - 317 с. - (Среднее профессиональное образование). - ISBN 978-5-16-015995-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1074164> (дата обращения: 03.06.2023). — Режим доступа: по подписке.

ПРИЛОЖЕНИЕ А

Список вопросов и ответов на тестовую программу

1. Элемент управления TextBox. Вопрос: Что понимают под КИС? Ответ дайте в именительном падеже, единственном числе. Ответ: корпоративная информационная система
2. Элемент управления: TextBox. Вопрос: Что в контексте подхода управления ИТ на основе услуг называется любым отклонением, любым событием, которое отличается от прописанных норм в договоре, от нормального состояния? Ответ: инцидент;
3. Элемент управления: CheckBox. Задание с выбором нескольких вариантов ответа. Вопрос: какие основные процессы принято описывать в рамках подхода управления ИТ на основе услуг? Варианты ответа: управление инцидентами, управление персоналом, управление проектом, управление изменениями, управление временем. Ответ: управление инцидентами, управление изменениями.
4. Элемент управления: CheckBox. Задание с выбором нескольких вариантов ответа. Вопрос: Что из перечисленного является стандартами корпоративных информационных систем? Варианты ответа: MPS, SCM, SCP, CSRP, DRP. Ответ: MPS, SCM, CSRP.
5. Элемент управления: RadioButton. Задание с выбором варианта ответа. Вопрос: Что из перечисленного является компонентом стандарта корпоративных систем ERP II? Варианты ответа: система управления конфигурациями, система бизнес-аналитики, система производственной

аналитики, система управления данными, система бизнес-логики. Ответ: система бизнес-аналитики.

6. Элемент управления: RadioButton. Задание с выбором варианта ответа. Вопрос: Что из перечисленного не является основной фазой проектирования КИС? Варианты ответа: Определение требований, реализация, верификация, эксплуатация и поддержка. Ответ: верификация.

7. Элемент управления: ListBox. Задание на соотнесение элементов друг к другу. Задание: Соотнесите сокращенные названия стандартов корпоративных информационных систем с их полным названием на русском языке. Левый столбец: 1. DRP, 2. ERP, 3. MPS, 4. MRP, 5. CRP. Правый столбец: 1. Управление календарным планированием, 2. Планирование материальных потребностей, 3. Управление загрузкой производственных мощностей, 4. Автоматизированные системы планирования потребностей распределения, 5. Планирование ресурсов предприятия. Ответ: 1-4, 2-5, 3-1, 4-2, 5-3.

8. Элемент управления: ListBox. Задание на соотнесение элементов друг к другу. Задание: Соотнесите сокращенные названия компонентов стандарта корпоративных информационных систем ERP II с их полным названием на русском языке. Левый столбец: 1. SRM, 2. SCM, 3. PLM, 4. Financials, 5. KM. Правый столбец: 1. Система управления знаниями о бизнесе, 2. Система управления финансами со стороны различных участников, 3. Система управления взаимоотношениями с поставщиками, 4. Система управления виртуальными логистическими цепочками, 5. Система управления жизненным циклом продукта. Ответ: 1-3, 2-4 3-5, 4-2, 5-1.

9. Элемент управления: ComboBox. Задание: Выберите названия процессов в рамках подхода ITIL соответственно их описанию. Описания процессов: 1. подразумевает решение любых событий, не являющихся частью

стандартных операций предоставления услуги, 2. предназначено для минимизации негативного влияния инцидентов на бизнес и уменьшение количества инцидентов, 3. используется для помощи управления инфраструктурой оборудования, 4. предназначено для согласования изменений, которые вносятся в ИТ-инфраструктуру предприятия. Элементы на выбор: 1. Управление проблемами, 2. Управление конфигурациями, 3. Управление инцидентами, 4. Управление релизами. Ответ: 1-3, 2-1, 3-2, 4-4.

10. Элемент управления: ComboBox. Задание: Выберите название параметров ИТ-сервиса соответственно их описанию. Описания параметров: 1. включает в себя функции, которые решает сервис, это задачи, которые определяются предметной областью использования данного сервиса, 2. это период времени, в течение которого ИТ подразделение поддерживает данный ИТ-сервис, то есть несёт ответственность за его непрерывное функционирование, 3. это доля согласованного времени обслуживания, в течение которого ИТ-сервис доступен для пользователей, 4. определяется средним временем наработки на отказ, то есть средним временем между двумя сбоями. Элементы на выбор: 1. Надежность, 2. Доступность, 3. Функциональность, 4. Время обслуживания. Ответ: 1-3, 2-4, 3-2, 4-1.

11. Элемент управления: TrackBar. Задание: В каких годах был разработан стандарт MRP II? Ответ: 1980-е.

12. Элемент управления: TrackBar. Задание: В каком году была выпущена ITIL Версия 3? Ответ: 2007.

13. Элемент управления: HScrollBar. Задание: Доступность ИТ-сервиса регламентирована в пределах 90-95%, режим работы составляет 24 часа 7 дней в неделю. Какое количество часов простоя сервиса в неделю допустимо? Ответ: от нуля до шестнадцати.

14. Элемент управления: HScrollBar. Задание: Доступность ИТ-сервиса регламентирована в пределах 90-95%, режим работы составляет 24 часа 7 дней в неделю. Каков показатель надежности сервиса? Ответ: от десяти до двадцати.

15. Элемент управления: CheckedListBox. Задание: Выберите характеристики ИТ-проекта. Варианты: 1. Изменчивость, 2. Ограниченнность персонала, 3. Ограниченность продолжительности, 4. Сложность, 5. Правовое и программное обеспечение, 6. Новизна для предметной области. Ответ: 1, 3, 5.

16. Элемент управления: CheckedListBox. Задание: Выберите основные компоненты стандарта Планирования потребности в мощностях. Варианты: 1. Логистические маршруты, 2. Главный календарный план производства, 3. Управление изменениями, 4. Система управления финансами со стороны различных участников, 5. Данные о рабочих центрах. Ответ: 2, 5.

ПРИЛОЖЕНИЕ В

Код программы

PlayerForm.cs

```
using AlgorithmsCourseworkLibrary;
using System;

using System.Windows.Forms;

namespace AlgorithmsCourseworkGUI
{
    public partial class PlayerForm : Form
    {
        OpenFileDialog openFileDialog;

        public PlayerForm()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
        }

        private void PlayerForm_Load(object sender, EventArgs e)
        {
            openFileDialog = new OpenFileDialog();
            this.Text = "Windows Media Player, версия = " + player.versionInfo;
        }

        private void ExitButton_Click(object sender, EventArgs e)
        {
```

```
        Application.Exit();
    }

    private void titleButton_Click(object sender, EventArgs e)
    {
        this.Hide();
        TitleForm nextForm = new TitleForm();
        nextForm.Show();
    }

    private void открытьToolStripMenuItem_Click(object sender, EventArgs e)
    {
        openFileDialog.ShowDialog();
        player.URL = openFileDialog.FileName;
        player.Ctlcontrols.play();
    }

    private void выходToolStripMenuItem_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void полноэкранныйРежимToolStripMenuItem_Click(object sender, EventArgs e)
    {
        if (player.playState == WMPLib.WMPPlayState.wmppsPlaying)
            player.fullScreen = true;
    }

    private void паузаToolStripMenuItem_Click(object sender, EventArgs e)
    {
        player.Ctlcontrols.pause();
```

```

    }

    private void начатьВоспроизведениеToolStripMenuItem_Click(object sender,
EventArgs e)
{
    player.Ctlcontrols.play();
}

private void выключитьвключитьЗвукToolStripMenuItem_Click(object sender,
EventArgs e)
{
    player.settings.mute = !player.settings.mute;
}

private void свойстваToolStripMenuItem_Click(object sender, EventArgs e)
{
    player.ShowPropertyPages();
}

}
}

```

Program.cs

```

using System;
using System.Windows.Forms;

namespace AlgorithmsCourseworkGUI
{
    internal static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new SplashScreenForm());
        }
    }
}

```

Question01.cs

```
using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question01 : Form
    {
        public Question01()
        {
            InitializeComponent();
            answerInput.Focus();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void confirmButton_Click(object sender, EventArgs e)
        {
            int result = AlgorithmsCoursework.UnaryQuestion(0, answerInput.Text,
                "корпоративная информационная система");
            if (result != -1)
            {

```

```

        this.Hide();

        Question02 nextForm = new Question02();
        nextForm.Show();
    }

}

}

}

```

Question02.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question02 : Form
    {
        public Question02()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}

```

```

private void confirmButton_Click(object sender, EventArgs e)
{
    int result = AlgorithmsCoursework.UnaryQuestion(1, answerInput.Text,
"инцидент");
    if (result != -1)
    {
        this.Hide();
        Question03 nextForm = new Question03();
        nextForm.Show();
    }
}
}

```

Question03.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question03 : Form
    {
        public Question03()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void confirmButton_Click(object sender, EventArgs e)
        {
            int result = AlgorithmsCoursework.Question03(checkBox1, checkBox2,
checkBox3, checkBox4, checkBox5);
            if (result != -1)
            {

```

```
        this.Hide();
        Question04 nextForm = new Question04();
        nextForm.Show();
    }
}
```

Question04.cs

```
using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question04 : Form
    {
        public Question04()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void confirmButton_Click(object sender, EventArgs e)
        {
            int result = AlgorithmsCoursework.Question04(checkBox1, checkBox2,
checkBox3, checkBox4, checkBox5);
            if (result != -1)
            {
                this.Hide();
                Question05 nextForm = new Question05();
                nextForm.Show();
            }
        }
    }
}
```

Question05.cs

```
using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question05 : Form
```

```

{
    public Question05()
    {
        InitializeComponent();

        AlgorithmsCoursework.SetupVersion(versionLabel);
        AlgorithmsCoursework.SetupTimer(timeLabel);
        AlgorithmsCoursework.SetupProgressBar(progressBar);
        AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
    }

    private void exitButton_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void confirmButton_Click(object sender, EventArgs e)
    {
        int result = AlgorithmsCoursework.Question05 radioButton1, radioButton2,
radioButton3, radioButton4, radioButton5);
        if (result != -1)
        {
            this.Hide();
            Question05 nextForm = new Question05();
            nextForm.Show();
        }
    }
}

```

Question06.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question06 : Form
    {
        public Question06()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void confirmButton_Click(object sender, EventArgs e)
        {

```

```

        int result = AlgorithmsCoursework.Question06 radioButton1, radioButton2,
radioButton3, radioButton4);
        if (result != -1)
        {
            this.Hide();
            Question06 nextForm = new Question06();
            nextForm.Show();
        }
    }
}
}

```

Question07.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question07 : Form
    {
        public Question07()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void confirmButton_Click(object sender, EventArgs e)
        {
            int result = AlgorithmsCoursework.Question07(
                listBox2.Items[0].ToString(), listBox2.Items[1].ToString(),
listBox2.Items[2].ToString(),
                listBox2.Items[3].ToString(), listBox2.Items[4].ToString()
            );
            this.Hide();
            Question08 nextForm = new Question08();
            nextForm.Show();
        }

        private void unionButton_Click(object sender, EventArgs e)
        {
            int i = listBox1.SelectedIndex;
            int j = listBox2.SelectedIndex;

            (listBox2.Items[i], listBox2.Items[j]) = (listBox2.Items[j],
listBox2.Items[i]);
            listBox2.SetSelected(i, true);
        }
    }
}

```

```
    }  
}
```

Question08.cs

```
using System;  
using System.Windows.Forms;  
  
using AlgorithmsCourseworkLibrary;  
  
namespace AlgorithmsCourseworkGUI  
{  
    public partial class Question08 : Form  
    {  
        public Question08()  
        {  
            InitializeComponent();  
  
            AlgorithmsCoursework.SetupVersion(versionLabel);  
            AlgorithmsCoursework.SetupTimer(timerLabel);  
            AlgorithmsCoursework.SetupProgressBar(progressBar);  
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);  
        }  
  
        private void exitButton_Click(object sender, EventArgs e)  
        {  
            Application.Exit();  
        }  
  
        private void confirmButton_Click(object sender, EventArgs e)  
        {  
            int result = AlgorithmsCoursework.Question08(  
                listBox2.Items[0].ToString(), listBox2.Items[1].ToString(),  
                listBox2.Items[2].ToString(),  
                listBox2.Items[3].ToString(), listBox2.Items[4].ToString()  
            );  
            this.Hide();  
            Question09 nextForm = new Question09();  
            nextForm.Show();  
        }  
  
        private void unionButton_Click(object sender, EventArgs e)  
        {  
            int i = listBox1.SelectedIndex;  
            int j = listBox2.SelectedIndex;  
  
            (listBox2.Items[i], listBox2.Items[j]) = (listBox2.Items[j],  
            listBox2.Items[i]);  
            listBox2.SetSelected(i, true);  
        }  
    }  
}
```

Question09.cs

```
using System;  
using System.Windows.Forms;
```

```

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question09 : Form
    {
        public Question09()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void confirmButton_Click(object sender, EventArgs e)
        {
            if (CB1.SelectedIndex == -1 || CB2.SelectedIndex == -1 ||
CB3.SelectedIndex == -1 || CB4.SelectedIndex == -1)
            {
                MessageBox.Show("Пожалуйста, выберите ответ во всех полях", "Нет
ответа", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }

            if (CB1.SelectedIndex == 2 && CB2.SelectedIndex == 0 && CB3.SelectedIndex
== 1 && CB4.SelectedIndex == 3)
            {
                MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                AlgorithmsCoursework.n += 1;
                AlgorithmsCoursework.answers[8] = true;
            }
            else
            {
                MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                AlgorithmsCoursework.answers[8] = false;
            }

            this.Hide();
            Question11 nextForm = new Question11();
            nextForm.Show();
        }
    }
}

```

Question10.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

```

```

namespace AlgorithmsCourseworkGUI
{
    public partial class Question10 : Form
    {
        public Question10()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void confirmButton_Click(object sender, EventArgs e)
        {
            if (CB1.SelectedIndex == -1 || CB2.SelectedIndex == -1 ||
CB3.SelectedIndex == -1 || CB4.SelectedIndex == -1)
            {
                MessageBox.Show("Пожалуйста, выберите ответ во всех полях", "Нет
ответа", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }

            if (CB1.SelectedIndex == 2 && CB2.SelectedIndex == 3 && CB3.SelectedIndex
== 1 && CB4.SelectedIndex == 0)
            {
                MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
                AlgorithmsCoursework.n += 1;
                AlgorithmsCoursework.answers[9] = true;
            }
            else
            {
                MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
                AlgorithmsCoursework.answers[9] = false;
            }

            this.Hide();
            Question11 nextForm = new Question11();
            nextForm.Show();
        }
    }
}

```

Question11.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

```

```

namespace AlgorithmsCourseworkGUI
{
    public partial class Question11 : Form
    {
        public Question11()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void confirmButton_Click(object sender, EventArgs e)
        {
            if (TB.Value == 2)
            {
                MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageButtons.OK, MessageBoxIcon.Information);
                AlgorithmsCoursework.n += 1;
                AlgorithmsCoursework.answers[10] = true;
            }
            else
            {
                MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageButtons.OK, MessageBoxIcon.Warning);
                AlgorithmsCoursework.answers[10] = false;
            }

            this.Hide();
            Question12 nextForm = new Question12();
            nextForm.Show();
        }
    }
}

```

Question12.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question12 : Form
    {
        public Question12()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);

```

```

        AlgorithmsCoursework.SetupProgressBar(progressBar);
        AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
    }

    private void exitButton_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void confirmButton_Click(object sender, EventArgs e)
    {
        if (TB.Value == 2007)
        {
            MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            AlgorithmsCoursework.n += 1;
            AlgorithmsCoursework.answers[11] = true;
        }
        else
        {
            MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            AlgorithmsCoursework.answers[11] = false;
        }

        this.Hide();
        Question13 nextForm = new Question13();
        nextForm.Show();
    }
}
}

```

Question13.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question13 : Form
    {
        private int value;

        public Question13()
        {
            InitializeComponent();
            value = HSB.Value;
            answerLabel.Text = "Ваш ответ: " + value.ToString();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {

```

```

        Application.Exit();
    }

    private void confirmButton_Click(object sender, EventArgs e)
    {
        if (HSB.Value >= 0 && HSB.Value <= 16)
        {
            MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            AlgorithmsCoursework.n += 1;
            AlgorithmsCoursework.answers[12] = true;
        }
        else
        {
            MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            AlgorithmsCoursework.answers[12] = false;
        }

        this.Hide();
        Question14 nextForm = new Question14();
        nextForm.Show();
    }

    private void HSB_Scroll(object sender, ScrollEventArgs e)
    {
        value = e.NewValue;
        answerLabel.Text = "Ваш ответ: " + value.ToString();
    }
}
}

```

Question14.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question14 : Form
    {
        private int value;

        public Question14()
        {
            InitializeComponent();
            value = HSB.Value;
            answerLabel.Text = "Ваш ответ: " + value.ToString();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {

```

```

        Application.Exit();
    }

    private void confirmButton_Click(object sender, EventArgs e)
    {
        if (HSB.Value >= 10 && HSB.Value <= 20)
        {
            MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            AlgorithmsCoursework.n += 1;
            AlgorithmsCoursework.answers[13] = true;
        }
        else
        {
            MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            AlgorithmsCoursework.answers[13] = false;
        }

        this.Hide();
        Question15 nextForm = new Question15();
        nextForm.Show();
    }

    private void HSB_Scroll(object sender, ScrollEventArgs e)
    {
        value = e.NewValue;
        answerLabel.Text = "Ваш ответ: " + value.ToString();
    }
}
}

```

Question15.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question15 : Form
    {
        public Question15()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void confirmButton_Click(object sender, EventArgs e)

```

```

    {
        if (CLB.CheckedItems.Count == 0)
        {
            MessageBox.Show("Вы не выбрали ни один из вариантов ответа", "Нет
ответа", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }

        if (
            CLB.CheckedItems.Count == 3 &&
            CLB.CheckedItems.Contains("Изменчивость") &&
            CLB.CheckedItems.Contains("Ограниченностя продолжительности") &&
            CLB.CheckedItems.Contains("Правовое и программное обеспечение")
        )
        {
            MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            AlgorithmsCoursework.n += 1;
            AlgorithmsCoursework.answers[14] = true;
        }
        else
        {
            MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            AlgorithmsCoursework.answers[14] = false;
        }

        this.Hide();
        Question16 nextForm = new Question16();
        nextForm.Show();
    }
}
}

```

Question16.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class Question16 : Form
    {
        public Question16()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
            AlgorithmsCoursework.SetupProgressBar(progressBar);
            AlgorithmsCoursework.SetupRemainingQuestionsCount(questionsCount);
        }

        private void exitButton_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }
    }
}

```

```

private void confirmButton_Click(object sender, EventArgs e)
{
    if (CLB.CheckedItems.Count == 0)
    {
        MessageBox.Show("Вы не выбрали ни один из вариантов ответа", "Нет
ответа", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    if (
        CLB.CheckedItems.Count == 2 &&
        CLB.CheckedItems.Contains("Главный календарный план производства") &&
CLB.CheckedItems.Contains("Данные о рабочих центрах")
    )
    {
        MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        AlgorithmsCoursework.n += 1;
        AlgorithmsCoursework.answers[15] = true;
    }
    else
    {
        MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        AlgorithmsCoursework.answers[15] = false;
    }

    this.Hide();
    ResultForm nextForm = new ResultForm();
    nextForm.Show();
}
}
}

```

ResultForm.cs

```

using System;
using System.Linq;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class ResultForm : Form
    {
        int[] sortedKeys = null;
        bool[] sortedValues = null;

        // ...
        public ResultForm()
        {
            InitializeComponent();

            gradeLabel.Text = "Ваша оценка по результатам тестирования: " +
AlgorithmsCoursework.CalculateGrade();
            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
        }
    }
}

```

```

    }

    private void exitButton_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void fetchButton_Click(object sender, EventArgs e)
    {
        int[,] resultsMatrix = AlgorithmsCoursework.GetResultsMatrix();
        AlgorithmsCoursework.OutputRows(resultView, resultsMatrix);
    }

    private void plotButton_Click(object sender, EventArgs e)
    {
        chart.Series[0].Points.AddXY(0, AlgorithmsCoursework.n);
        chart.Series[1].Points.AddXY(1, AlgorithmsCoursework.answers.Length - AlgorithmsCoursework.n);
    }

    private void writeButton_Click(object sender, EventArgs e)
    {
        if (resultView.RowCount == 0)
        {
            MessageBox.Show("Получите результаты тестирования", "Операция невозможна", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }

        AlgorithmsCoursework.WriteWord(resultView);
    }

    private void openButton_Click(object sender, EventArgs e)
    {
        AlgorithmsCoursework.OpenFile("Microsoft Word (*.docx)|*.docx|All files (*.*)|*.*",
            "Выберите документ Word");
    }

    private void writeExcelButton_Click(object sender, EventArgs e)
    {
        AlgorithmsCoursework.WriteExcel(
            Enumerable.Range(1, AlgorithmsCoursework.answers.Length).ToArray(),
            AlgorithmsCoursework.answers
        );
    }

    private void openExcelButton_Click(object sender, EventArgs e)
    {
        AlgorithmsCoursework.OpenFile(
            "Excel Workbook (*.xlsx)|*.xlsx|Excel Macro-Enabled Workbook (*.xlsm)|*.xlsm|All files (*.*)|*.*",
            "Выберите документ Excel"
        );
    }

    private void sortButton_Click(object sender, EventArgs e)
    {
        sortedKeys = new int[AlgorithmsCoursework.answers.Length];
        sortedValues = new bool[AlgorithmsCoursework.answers.Length];
    }
}

```

```

        for (int i = 0; i != AlgorithmsCoursework.answers.Length; ++i)
        {
            sortedKeys[i] = i + 1;
            sortedValues[i] = AlgorithmsCoursework.answers[i];
        }

        AlgorithmsCoursework.Sort(sortedKeys, sortedValues);
        AlgorithmsCoursework.OutputRows(sortedView, sortedKeys, sortedValues);
    }

    private void writeSortExcelButton_Click(object sender, EventArgs e)
    {
        if (sortedKeys == null || sortedValues == null)
        {
            MessageBox.Show("Выполните сортировку", "Операция невозможна",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            return;
        }

        AlgorithmsCoursework.Sort(sortedKeys, sortedValues);
        AlgorithmsCoursework.WriteExcel(sortedKeys, sortedValues);
    }

    private void insertColumnButton_Click(object sender, EventArgs e)
    {
        string name = "";
        while (name == "")
            name = Microsoft.VisualBasic.Interaction.InputBox("Пожалуйста, введите
название нового столбца", "Создание столбца");

        var column = new DataGridViewTextBoxColumn();
        column.ReadOnly = false;
        column.Name = name;
        resultView.Columns.Add(column);
    }

    private void titleButton_Click(object sender, EventArgs e)
    {
        AlgorithmsCoursework.ClearResults();

        this.Hide();
        TitleForm nextForm = new TitleForm();
        nextForm.Show();
    }
}

```

SplashScreenForm.cs

```

using System;
using System.Windows.Forms;

namespace AlgorithmsCourseworkGUI
{
    public partial class SplashScreenForm : Form
    {
        Timer timer = new Timer();

```

```

    public SplashScreenForm()
    {
        InitializeComponent();

        // Устанавливаем таймер, чтобы через пять секунд он выполнил указанную
функцию
        timer.Tick += SwitchToTitleForm;
        timer.Interval = 5000;
        timer.Start();
    }

    public void SwitchToTitleForm(object e, EventArgs v)
    {
        // Останавливаем таймер. Иначе он начнет повторяться. Он нам больше не
нужен
        timer.Stop();

        // Переходим к титульной форме
        this.Hide();
        TitleForm nextForm = new TitleForm();
        nextForm.Show();
    }
}

```

TheoryForm.cs

```

using AlgorithmsCourseworkLibrary;
using Microsoft.Web.WebView2.Core;
using System;
using System.IO;
using System.Windows.Forms;

namespace AlgorithmsCourseworkGUI
{
    public partial class TheoryForm : Form
    {
        public TheoryForm()
        {
            InitializeComponent();
            InitializeAsync();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);
        }

        private async void InitializeAsync()
        {
            var env = await CoreWebView2Environment.CreateAsync(userDataFolder:
Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData));
            await webView.EnsureCoreWebView2Async(env);

            webView.CoreWebView2.SetVirtualHostNameToFolderMapping(hostName:
"localhost",
                folderPath: "", accessKind:
CoreWebView2HostResourceAccessKind.Allow);
        }
    }
}

```

```

        webView.CoreWebView2.Navigate("http://localhost/report.pdf");

    }

    private void SetPage(int pageIndex)
    {
        string newUri = "http://localhost/report.pdf#page=" + pageIndex;
        webView.CoreWebView2.Navigate(newUri);
    }

    private void ExitButton_Click(object sender, EventArgs e)
    {
        Application.Exit();
    }

    private void TitleButton_Click(object sender, EventArgs e)
    {
        this.Hide();
        TitleForm nextForm = new TitleForm();
        nextForm.Show();
    }

    private void HandleItem_Click(object sender, EventArgs e)
    {
        String tag = (String)((ToolStripItem)sender).Tag;
        if (tag != null && tag.Length != 0) {
            SetPage(Convert.ToInt32(tag));
        }
    }
}
}

```

TitleForm.cs

```

using System;
using System.Windows.Forms;

using AlgorithmsCourseworkLibrary;

namespace AlgorithmsCourseworkGUI
{
    public partial class TitleForm : Form
    {
        public TitleForm()
        {
            InitializeComponent();

            AlgorithmsCoursework.SetupVersion(versionLabel);
            AlgorithmsCoursework.SetupTimer(timeLabel);

            KeyPreview = true;
        }
        private void exitButton_Click(object sender, System.EventArgs e)
        {
            Application.Exit();
        }

        private void theoryButton_Click(object sender, System.EventArgs e)
        {

```

```

        this.Hide();
        TheoryForm nextForm = new TheoryForm();
        nextForm.Show();
    }

    private void testButton_Click(object sender, System.EventArgs e)
    {
        this.Hide();
        Question01 nextForm = new Question01();
        nextForm.Show();
    }

    private void playerButton_Click(object sender, EventArgs e)
    {
        this.Hide();
        PlayerForm nextForm = new PlayerForm();
        nextForm.Show();
    }

    private void resultButton_Click(object sender, EventArgs e)
    {
        this.Hide();
        ResultForm nextForm = new ResultForm();
        nextForm.Show();
    }

    private void TitleForm_KeyDown(object sender, KeyEventArgs e)
    {
        if (e.Alt && e.KeyCode == Keys.F5)
        {
            MessageBox.Show("Режим разработчика успешно активирован");
            resultButton.Visible = true;
            e.Handled = true;
        }
    }
}
}

```

AlgorithmsCoursework.cs

```

using System;
using System.Drawing;
using System.Windows.Forms;

using Microsoft.Office.Interop.Excel;
using Microsoft.Office.Interop.Word;
using CheckBox = System.Windows.Forms.CheckBox;

namespace AlgorithmsCourseworkLibrary
{
    public class AlgorithmsCoursework
    {
        public static int n = 0;
        public static bool[] answers = new bool[16];
        public static int version = 100;

        public static int UnaryQuestion(int index, string userAnswer, string
rightAnswer)
        {

```

```

        userAnswer = userAnswer.Trim().ToLower();
        if (userAnswer == "")
        {
            MessageBox.Show("Вы не ввели ответ на вопрос", "Нет ответа",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            return -1;
        }
        else if (userAnswer == rightAnswer)
        {
            MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
            n += 1;
            answers[index] = true;
            return 1;
        }
        else
        {
            MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            answers[index] = false;
            return 0;
        }
    }

    public static int Question03(CheckBox CB1, CheckBox CB2, CheckBox CB3,
CheckBox CB4, CheckBox CB5)
{
    if (!CB1.Checked && !CB2.Checked && !CB3.Checked && !CB4.Checked && !
CB5.Checked)
    {
        MessageBox.Show("Вы не выбрали ни один из вариантов ответа", "Нет
ответа", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return -1;
    }
    else if (CB1.Checked && !CB2.Checked && !CB3.Checked && CB4.Checked && !
CB5.Checked)
    {
        MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        n += 1;
        answers[2] = true;
        return 1;
    }
    else
    {
        MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        answers[2] = false;
        return 0;
    }
}

public static int Question04(CheckBox CB1, CheckBox CB2, CheckBox CB3,
CheckBox CB4, CheckBox CB5)
{
    if (!CB1.Checked && !CB2.Checked && !CB3.Checked && !CB4.Checked && !
CB5.Checked)
    {
        MessageBox.Show("Вы не выбрали ни один из вариантов ответа", "Нет
ответа", MessageBoxButtons.OK, MessageBoxIcon.Error);

```

```

        return -1;
    }
    else if (CB1.Checked && CB2.Checked && !CB3.Checked && CB4.Checked && !
CB5.Checked)
    {
        MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        n += 1;
        answers[3] = true;
        return 1;
    }
    else
    {
        MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        answers[3] = false;
        return 0;
    }
}

public static int Question05(RadioButton RB1, RadioButton RB2, RadioButton
RB3, RadioButton RB4, RadioButton RB5)
{
    if (!RB1.Checked && !RB2.Checked && !RB3.Checked && !RB4.Checked && !
RB5.Checked)
    {
        MessageBox.Show("Вы не выбрали ни один из вариантов ответа", "Нет
ответа", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return -1;
    }
    else if (RB2.Checked)
    {
        MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        n += 1;
        answers[4] = true;
        return 1;
    }
    else
    {
        MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        answers[4] = false;
        return 0;
    }
}

public static int Question06(RadioButton RB1, RadioButton RB2, RadioButton
RB3, RadioButton RB4)
{
    if (!RB1.Checked && !RB2.Checked && !RB3.Checked && !RB4.Checked)
    {
        MessageBox.Show("Вы не выбрали ни один из вариантов ответа", "Нет
ответа", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return -1;
    }
    else if (RB3.Checked)
    {

```

```

        MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        n += 1;
        answers[5] = true;
        return 1;
    }
    else
    {
        MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        answers[5] = false;
        return 0;
    }
}

public static int Question07(string answer1, string answer2, string answer3,
string answer4, string answer5)
{
    if (
        answer1 == "Автоматизированные системы планирования потребностей
распределения" && answer2 == "Планирование ресурсов предприятия" &&
        answer3 == "Управление календарным планированием" && answer4 ==
"Планирование материальных потребностей" &&
        answer5 == "Управление загрузкой производственных мощностей"
    )
    {
        MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        n += 1;
        answers[6] = true;
        return 1;
    }
    else
    {
        MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        answers[6] = false;
        return 0;
    }
}

public static int Question08(string answer1, string answer2, string answer3,
string answer4, string answer5)
{
    if (
        answer1 == "Система управления взаимоотношениями с поставщиками" &&
        answer2 == "Система управления виртуальными логистическими цепочками"
&&
        answer3 == "Система управления жизненным циклом продукта" &&
        answer4 == "Система управления финансами со стороны различных
участников" &&
        answer5 == "Система управления знаниями о бизнесе"
    )
    {
        MessageBox.Show("Вы ответили правильно", "Правильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Information);
        n += 1;
        answers[7] = true;
        return 1;
    }
}

```

```

        }
        else
        {
            MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
            answers[7] = false;
            return 0;
        }
    }

    public static void SetupVersion(ToolStripStatusLabel label)
{
    int version = AlgorithmsCoursework.version;
    int major = version / 100, minor = (version / 10) % 10, patch = version %
10;
    label.Text = string.Format("Версия программы: {0}.{1}.{2}", major, minor,
patch);
}

    public static void SetupTimer(ToolStripStatusLabel label)
{
    Timer timer = new Timer();
    timer.Tick += (object e, EventArgs v) => UpdateDate(label);
    timer.Interval = 1000;
    timer.Start();
    UpdateDate(label);
}

    public static void SetupProgressBar(ProgressBar progressBar)
{
    progressBar.Minimum = 0;
    progressBar.Maximum = 16;
    progressBar.Value = n;
}

    public static void SetupRemainingQuestionsCount(System.Windows.Forms.Label
label)
{
    label.Text = "Осталось " + (answers.Length - n) + " вопросов";
}

    private static void UpdateDate(ToolStripStatusLabel label)
{
    string dateInfo = DateTime.Now.ToString("ddd, dd MMMM yyyy, HH:mm:ss");
    label.Text = dateInfo.Substring(0, 1).ToUpper() + dateInfo.Substring(1);
}

// ResultForm.cs
public static int CalculateGrade()
{
    double correctProcent = (double) n / answers.Length;
    if (correctProcent >= 100)
        return 5;
    else if (correctProcent >= 75)
        return 4;
    else if (correctProcent >= 60)
        return 3;
    return 2;
}

```

```

    }

    public static int[,] GetResultsMatrix()
    {
        int[,] resultsMatrix = new int[answers.Length, 2];
        for (int i = 0; i != answers.Length; i++)
        {
            resultsMatrix[i, 0] = i + 1;
            resultsMatrix[i, 1] = answers[i] == true ? 1 : 0;
        }
        return resultsMatrix;
    }

    public static void OutputRows(DataGridView view, int[,] resultsMatrix)
    {
        view.Rows.Clear();
        for (int i = 0; i != resultsMatrix.GetLength(0); ++i)
        {
            view.Rows.Add(resultsMatrix[i, 0], resultsMatrix[i, 1]);
        }
    }

    public static void OutputRows(DataGridView view, int[] keys, bool[] values)
    {
        view.Rows.Clear();
        for (int i = 0; i != keys.Length; i++)
        {
            view.Rows.Add(keys[i], values[i] == true ? "Верно" : "Неверно");
        }
    }

    public static void SaveChartAsImage(ChartObject chart)
    {
        chart.CopyPicture(
            Microsoft.Office.Interop.Excel.XlPictureAppearance.xlScreen,
            Microsoft.Office.Interop.Excel.XlCopyPictureFormat.xlBitmap
        );
        Bitmap image = new Bitmap(Clipboard.GetImage());

        SaveFileDialog saveFileDialog = new SaveFileDialog
        {
            Filter = "Image (*.jpg) | *.jpg",
            Title = "Сохранить изображение"
        };

        while (saveFileDialog.ShowDialog() != DialogResult.OK)
            continue;

        image.Save(saveFileDialog.FileName,
        System.Drawing.Imaging.ImageFormat.Jpeg);
    }

    public static void WriteExcel(int[] keys, bool[] values)
    {
        Microsoft.Office.Interop.Excel.Application app = new
        Microsoft.Office.Interop.Excel.Application();
        Workbook workBook;
        Worksheet workSheet;
    }
}

```

```

workBook = app.Workbooks.Add();
workSheet = (Worksheet)workBook.Worksheets.get_Item(1);
workSheet.Name = "Массив исходный";
workSheet.Cells[1, 1] = "Массив ответов";
for (int i = 0; i != values.Length; i++)
{
    workSheet.Cells[2, i + 1] = "[" + keys[i] + "]";
    workSheet.Cells[3, i + 1] = Convert.ToInt32(values[i]);
}

Microsoft.Office.Interop.Excel.Range range =
workSheet.Range[workSheet.Cells[2, 1], workSheet.Cells[3, values.Length]];
range.Cells.Font.Name = "Times New Roman";
range.Cells.Font.Size = 14;
range.Cells.Columns.AutoFit();
range.Borders.get_Item(XLBordersIndex.xlEdgeBottom).LineStyle =
    Microsoft.Office.Interop.Excel.XLLineStyle.xlContinuous;
range.Borders.get_Item(XLBordersIndex.xlEdgeRight).LineStyle =
    Microsoft.Office.Interop.Excel.XLLineStyle.xlContinuous;
range.Borders.get_Item(XLBordersIndex.xlInsideHorizontal).LineStyle =
    Microsoft.Office.Interop.Excel.XLLineStyle.xlContinuous;
range.Borders.get_Item(XLBordersIndex.xlInsideVertical).LineStyle =
    Microsoft.Office.Interop.Excel.XLLineStyle.xlContinuous;
range.Borders.get_Item(XLBordersIndex.xlEdgeTop).LineStyle =
    Microsoft.Office.Interop.Excel.XLLineStyle.xlContinuous;

workSheet.Cells[4, 1] = "Правильные ответы";
workSheet.Cells[4, 2] = "=SUM(A3:P3)";
workSheet.Cells[5, 1] = "Неправильные ответы";
workSheet.Cells[5, 2] = "=" + values.Length + "-SUM(A3:P3)";

object misValue = System.Reflection.Missing.Value;

Microsoft.Office.Interop.Excel.Range chartRange;

ChartObjects xlCharts =
(ChartObjects)workSheet.ChartObjects(Type.Missing);
ChartObject myChart = (ChartObject)xlCharts.Add(0, 80, 300, 250);
Microsoft.Office.Interop.Chart chartPage = myChart.Chart;

chartRange = workSheet.get_Range("A4", "B5");
chartPage.SetSourceData(chartRange, misValue);
chartPage.ChartType = XlChartType.xlColumnClustered;
chartPage.Legend.Delete();

SaveChartAsImage(myChart);

workSheet.Range[("A7")].Select();
app.Visible = true;
app.UserControl = true;
}
public static void WriteWord(DataGridView view)
{
    Microsoft.Office.Interop.Word.Application app = new
Microsoft.Office.Interop.Word.Application();
    var Word = new Microsoft.Office.Interop.Word.Application
    {
        Visible = true,

```

```

};

var inf = Type.Missing;
var Document = Word.Documents.Add(inf);
Word.Selection.TypeText("Массив ответов");

object t1 = WdDefaultTableBehavior.wdWord9TableBehavior;
object t2 = WdAutoFitBehavior.wdAutoFitContent;

int length = view.RowCount, columns = view.ColumnCount;
Table tbl = Word.ActiveDocument.Tables.Add(Word.Selection.Range, columns,
length, t1, t2);

for (int i = 0; i != columns; ++i)
{
    for (int j = 0; j != length; ++j)
    {
        object value = view.Rows[j].Cells[i].Value;
        tbl.Cell(i + 1, j + 1).Range.InsertAfter(value == null ? "null":
value.ToString());
    }
}

public static void Sort(int[] keys, bool[] values)
{
    for (int j = 1; j != values.Length; ++j)
    {
        var key = keys[j];
        var value = values[j];
        var i = j - 1;

        while (i >= 0 && Convert.ToInt32(values[i]) > Convert.ToInt32(value))
        {
            keys[i + 1] = keys[i];
            values[i + 1] = values[i];
            i--;
        }

        keys[i + 1] = key;
        values[i + 1] = value;
    }
}

public static void OpenFile(string filters, string title)
{
    OpenFileDialog openFileDialog = new OpenFileDialog
    {
        Filter = filters,
        Title = title
    };
    if (openFileDialog.ShowDialog() != DialogResult.OK)
    {
        MessageBox.Show("Вы не выбрали файл", "Открыть", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        return;
    }
    System.Diagnostics.Process.Start(openFileDialog.FileName);
}

```

```
// Сброс состояния прохождения тестирования к изначальному
public static void ClearResults()
{
    n = 0;
    answers = new bool[16];
}
}
```