Министерство цифрового развития, связи и массовых коммуникаций РФ

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»

Тема: «Моделирование потоков данных в информационных системах»

Выполнил:

студент группы БФИ2204

Петрушин Е.И.

Научный руководитель:

доц. кафедры «Информатика»

к. п. н. Гуриков С.Р.

Москва, 2023 г.

Оглавление

[**ВВЕДЕНИЕ** 3](#_Toc136755534)

[ТЕХНИЧЕСКОЕ ЗАДАНИЕ 5](#_Toc136755535)

[ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 11](#_Toc136755549)

[1.1 Основные понятия теории информационных процессов и систем. 12](#_Toc136755550)

[1.2 Модели жизненного цикла информационных систем (ИС). 15](#_Toc136755551)

[1.3 Автоматизированные информационные системы (АИС) и информационное моделировании. 22](#_Toc136755552)

[ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ 29](#_Toc136755553)

[2.1. Разработка электронного конспекта 29](#_Toc136755554)

[2.2. Разработка тестовой программы 31](#_Toc136755555)

[На данном этапе перейдем к описанию итоговой формы, которая представлена на рисунках 18-19. 43](#_Toc136755556)

[2.3 Описание дополнительных возможностей 44](#_Toc136755557)

[**ЗАКЛЮЧЕНИЕ** 47](#_Toc136755558)

[Список используемых источников 48](#_Toc136755559)

[Приложение А 48](#_Toc136755560)

[Приложение B 51](#_Toc136755561)

**ВВЕДЕНИЕ**

В современный период развития процесс эффективного проектирования информационных систем становится неотъемлемой частью жизнедеятельности организации в целом. Руководство компаний так же понимает, что в настоящее время успешность и прибыльность зачастую зависит и от уровня развития используемых информационных технологий, и от уровня развития профессионализма сотрудников в сфере использования информационных технологий. Моделирование потоков данных в информационных системах — это процесс документирования движения данных внутри системы, от их создания до их потребления, с целью идентификации, анализа и оптимизации потока информации. Это включает в себя идентификацию источников данных, применяемых к ним преобразований и мест назначения, где они хранятся или используются.

В прошлом, методология моделирования информационных систем была положена в основу CASE-технологии. В свою очередь, в основу этой технологии легла серия стандартов обозначаемых аббревиатурой IDEF. Затем на рынке программных продуктов появились средства для проведения серьезного анализа в структурных методах. Таким образом, CASE-технология—это инструмент для системных аналитиков и программистов, позволяющий автоматизировать процессы анализа, проектирования и реализации информационных систем.

Прежде всего, создание CASE-технологий, которые во много сокращают сроки проектирования информационных систем, позволяют организовать одновременную коллективную работу, оперативно вносить изменения и быстро реагировать на изменение обстановки на предприятии. Особое место занимают современные информационные технологии ведения электронной коммерции, работа с заказчиками и поставщиками.

В целом, моделирование потоков данных в информационных системах включает в себя **особую актуальность** проектирования и внедрения эффективных и действенных систем, которые могут поддерживать цели и задачи организации.

**Объектом исследования** является оптимизация производительности в области моделирования потоков данных в информационных системах.

**Предметом исследования** является разработка новых методов моделирования потока данных. Например, можно разработать новые методы визуализации или алгоритмы машинного обучения для выявления узких мест в потоке данных.

**Целью работы является** анализ теоретического материала и разработка тестовой программы по теме исследования “ Моделирование потоков данных в информационных системах”, использование которой студент получит определенные знания в данной области.

Поставленная цель определила следующие основные задачи исследования:

1. Сделать анализ всего теоретического материала по предложенной теме – “ Моделирование потоков данных в информационных системах”.

2. Разработать программное средство для проверки знаний по пройденному материалу.

**Методы исследования.** Для решения поставленных задач были использованы теоретические методы исследования. Теоретическую основу исследования составили труды в области моделирования потоков данных в информационных системах В. Ю. Белаш, Н.В. Тимошина.

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

1. **Введение**

Областью применения разрабатываемого программного продукта является углубленное изучение, а также контроль качества усвоения студентами основных определений, принципов и особенностей, используемых при построении моделирование потоков данных информационных систем в основе CASE-технологии.

# Основание для разработки

* лист утверждения тем курсовых работ (протокол №5 заседания кафедры от 01.02.2023 г.), подписанный научным руководителем: доцентом кафедры «Информатика» Гуриковым С.Р.;
* наименование разработки  «Моделирование потоков данных в информационных системах».

# Назначение разработки

Разрабатываемый программный продукт предназначен для изучения и, в последующем, контроля качества усвоения студентами основных определений, принципов и методов, используемых при изучении моделирования потоков данных в информационных системах.

# Требования к программе или программному изделию

* 1. **Требования к функциональным характеристикам**
     + осуществить вывод ответов теста в dataGridView на итоговой форме
     + осуществить построение диаграммы, отображающей результаты тестирования на основе использования элемента управления Chart
     + Упорядочить результаты теста, представленные в виде одномерного массива в другом компоненте dataGridView, с использованием метода сортировки бинарными вставками, при этом используется алгоритм сортировки. Отсортированный массив записать в табличный процессор Microsoft Excel. Создать макрос для форматирования таблиц и установки границ в таблицах, а также создать кнопку на листе Excel, которая выполняет написанный макрос.
     + Упорядочить результаты теста, представленные в виде одномерного массива в другом компоненте dataGridView, с использованием метода сортировки простой вставкой, при этом используется алгоритм сортировки. 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. Создать меню и осуществить реализацию программного кода, который вызывает появление окна OpenFileDialog с последующим открытием созданного файла.

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.

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.

# 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.

# Условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям, предъявляемым 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.

# Требования к составу и параметрам технических средств

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

* процессор с тактовой частотой не ниже 1 Ггц;
* оперативная память объемом не менее 2 Гб;
* периферийные устройства: клавиатура, мышь;
* монитор с разрешающей способностью не ниже 1280\*720 px;
* жесткий диск объемом не менее 2 Гб;
* желательно наличие принтера для печати отчета итогов работы программы.

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 целесообразнее использовать операционную систему семейства Windows не ниже Windows 8.

# 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 Visual С#. В качестве интегрированной среды разработки программы должна быть использована среда MS Visual Studio.

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

# Требования к маркировке и упаковке

Программа поставляется в виде программного изделия - 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 укладывают 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.

# 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.

# Требования к программной документации

Предварительный перечень программной документации:

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

# Технико-экономические показатели

В данной работе не рассчитываются.

# Стадии и этапы разработки

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

# Порядок контроля и приемки

* контроль и приемка программного продукта осуществляется в течение семестра поэтапно в соответствии с план-графиком;
* КР в окончательном виде предоставляется на проверку преподавателям за неделю до ее защиты;
* курсовая работа подлежит защите, в ходе которой студент представляет свою разработку и выполняет практические задания с целью демонстрации практических навыков на компьютере. На защите КР преподавателями не осуществляется проверка программной разработки и пояснительной записки;
* в случае регулярного и качественного выполнения план-графика сдачи этапов КР, студент может быть освобожден от защиты КР;
* в случае отсутствия в пояснительной записке глав или подпунктов, которые были оговорены в техническом задании, студенту на защите КР выставляется оценка «неудовлетворительно»;
* в случае невыполнения план-графика в течение семестра, студенту будут выданы дополнительные задания;
* в случае обнаружения заимствования чужих курсовых работ, студенту выставляется оценка «неудовлетворительно»;
* оценку программного продукта производят преподаватели, осуществляющие проверку курсовых работ.

# ГЛАВА 1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## 1.1 Основные понятия теории информационных процессов и систем.

Первый вариант общей теории систем был предложен в 1912 г. А.А. Богдановым в виде учения, названного **технологией**.

Тектология Богданова — это общая теория организации и дезорганизации, наука об универсальных типах и закономерностях структурного преобразования любых систем.

А.А. Богданов одним из первых в мире ввёл понятие системности. Учёный разработал идею о структурной устойчивости системы и её условиях.

В самой системе, по мнению Богданова, выделяется два вида закономерностей:

а) формирующие, т.е. закономерности развития, приводящие к переходу системы в другое качество;

б) регулирующие, т.е. закономерности функционирования, способствующие стабилизации нынешнего качества системы.

Богданов ввёл также ряд понятий, характеризующих этапы развития различных систем:

* «комплексия» употреблялся им для обозначения ситуации, когда система представляет собой чисто механическое объединение элементов, между которыми ещё не начались процессы взаимодействия;
* «конъюгация» означает такой этап развития системы, когда начинается сотрудничество между отдельными элементами системы;
* «ингрессия» выражает этап перехода системы к новому качеству;
* «дезингрессия», наоборот, означает процесс деградации системы, её распада как целостного объединения.

Продолжил работу А.А. Богданова австрийский биолог и философ Людвиг фон Берталанфи (1901–1972). Он создал второй вариант общей теории систем.

Термин «Система»:

* Система — это теория;
* Система — это классификация;
* Система — это завершённый метод практической деятельности;
* Система — некоторый способ мыслительной деятельности;
* Система — это совокупность объектов природы;
* Система — это некоторое явление общества;
* Система — это совокупность установившихся норм жизни, правил поведения.

Людвиг фон Берталанфи определял систему как комплекс взаимодействующих элементов или как совокупность элементов, находящихся в определённых отношениях друг с другом и со средой.

Подходы к определению системы:

а) Дескриптовый подходзаключается в том, что характер функционирования системы объясняют её структурой, элементами;

б) Констурктивый подход по заданной функции конструируется соответствующая ей структура. При этом система должна соответствовать некоторым целям конструирования.

Свойства (закономерности) систем:

* ограниченность;
* целостность;
* интегративность;
* коммуникативность;
* историчность;
* закон необходимого разнообразия;
* закономерность осуществимости и потенциальной эффективности систем;
* закономерность целеобразования;
* зависимость представления о цели и формулировки цели от стадии познания объекта (процесса);
* зависимость цели от внутренних и внешних факторов;
* возможность сведения задачи формирования общей цели к задаче структуризации цели;
* зависимость способа представления структуры целей от стадии познания объекта или процесса (продолжение первой закономерности).

Теория информационных процессов — совокупность моделей и методов, предназначенных (или используемых) для анализа информационных процессов (ИП), происходящих в технических, экономических, социальных, биологических, экологических и других системах.

**Информационные технологии, в широком понимании** — это разработка алгоритмов, программ и их применение на компьютере с целью анализа и синтеза сложных систем (технических, информационных, биологических, экономических, социальных и др.).

Согласно определению, принятому ЮНЕСКО, **Информационные технологии** — это комплекс взаимосвязанных, научных, технологических, инженерных дисциплин, изучающих:

* методы эффективной организации труда людей, занятых обработкой и хранением информации;
* вычислительную технику и методы организации и взаимодействия с людьми и производственным оборудованием;
* их практические приложения, а также связанные со всем этим социальные, экономические и культурные проблемы.

**Информационная система (ИС)** — это организационно-упорядоченная взаимосвязанная совокупность средств и методов ИТ, используемых для хранения, обработки и выдачи информации в интересах достижения поставленной цели.

## 1.2 Модели жизненного цикла информационных систем (ИС).

**Жизненный цикл информационной системы** — это непрерывный процесс, начинающийся с момента принятия решения о создании информационной системы и заканчивающийся в момент полного изъятия ее из эксплуатации.

Согласно стандарту структура жизненного цикла основывается на трех группах процессов:

* основные процессы (заказ, поставка, разработка, эксплуатация, сопровождение);
* вспомогательные процессы (обеспечивают выполнение основных процессов):
* документирование— работы по разработке, выпуску, редактированию, распространению и сопровождению документов, в которых нуждаются все заинтересованные лица;
* управление конфигурацией (конфигурационное управление) включает работы: определение и установление состояния программных объектов в системе; управление изменениями и выпуском объектов; обеспечение полноты, совместимости и правильности объектов; управление хранением, обращением и поставкой объектов;
* обеспечение качества — работы по обеспечению соответствия создаваемой системы и реализуемых процессов жизненного цикла установленным требованиям и утвержденным планам;
* верификация— работы соответствующего субъекта (заказчика, поставщика или независимой стороны) по проверке соответствия создаваемых промежуточных результатов установленным требованиям по мере реализации проекта. Различают верификацию договора, процесса, требований, проекта, системы, сборки системы и документации;
* аттестация— работы соответствующего субъекта по проверке полного соответствия требований и конечного продукта функциональному назначению системы;
* совместный анализ— работы по оценке состояния или результатов какой-либо работы (системы);
* аудит— работы независимых (по отношению к проекту) экспертов по определению соответствия деятельности субъекта принятым требованиям, планам и условиям договора;
* разрешение проблем— работы по анализу и устранению проблем, обнаруженных при реализации проекта;
* организационные процесс**;**
* управление проектами— работы по планированию и управлению процессами, включая контроль, проверку и оценку выполненных работ с формированием отчетности;
* создание инфраструктуры проекта — работы по установлению и обеспечению инфраструктуры, необходимой для любого другого процесса. Инфраструктура может содержать технические и программные средства, инструментальные средства, методики, стандарты и условия для разработки, эксплуатации или сопровождения системы;
* усовершенствование— работы по оценке, контролю и улучшению процессов жизненного цикла;
* обучение— работы по планированию и проведению обучения персонала, включая разработку учебных материалов. При этом под персоналом понимаются не только конечные пользователи, которые будут эксплуатировать систему, но и разработчики системы. Согласно стандарту структура жизненного цикла основывается на трех группах процессов.

Стадии жизненного цикла информационной системы**:**

1. **Системный анализ:** обследование объекта и обоснование необходимости создания ИС; формирование требований заказчика к ИС; оформление договора между разработчиком и заказчиком.
2. **Анализ требований**: поиск путей удовлетворения требований заказчика на уровне концепции создаваемой системы (структура, функции, программно-техническая платформа, режимы); рассмотрение альтернативных вариантов концепции системы, их анализ и выбор лучшей концепции.
3. **Проектирование:** разработка, согласование и утверждение технического задания на создание ИС; разработка предварительных проектных решений; разработка проектных решений по системе и ее частям.
4. **Кодирование (реализация):** разработка рабочей документации на систему и ее части; разработка программных и технических средств и/или адаптация приобретаемых; тестирование средств.
5. **Тестирование:** загрузка базы данных типовыми исходными данными и тестами; интеграция программ и тестирование в имитированной среде; интеграция программных средств с аппаратными в реальной операционной и внешней среде; тестирование в реальной среде; разработка комплекта документации для пользователей.
6. **Внедрение и сопровождение**: подготовка объекта автоматизации к вводу ИС в действие; обучение и консультации пользователей; поставка программного обеспечения и документации на объекты внедрения.

**Модели жизненного цикла информационной системы**

**Каскадная модель** подразумевает линейную последовательность выполнения стадий создания информационной системы. Переход с одной стадии на следующую происходит только после того, как будет полностью завершена работа на текущей.

На рисунке 1 представлен снимок каскадной модели.



Рисунок 1 – Каскадная модель

Данная модель применяется при разработке информационных систем, для которых в самом начале разработки можно достаточно точно и полно сформулировать все требования.

Достоинства модели:

* на каждой стадии формируется законченный набор документации, программного и аппаратного обеспечения, отвечающий критериям полноты и согласованности;
* выполняемые в четкой последовательности стадии позволяют уверенно планировать сроки выполнения работ и соответствующие ресурсы (денежные, материальные и людские).

Недостатки модели:

* реальный процесс разработки информационной системы редко полностью укладывается в такую жесткую схему. Особенно это относится к разработке нетиповых и новаторских систем;
* жизненный цикл основан на точной формулировке исходных требований к информационной системе. Реально в начале проекта требования заказчика определены лишь частично;
* основной недостаток — результаты разработки доступны заказчику только в конце проекта. В случае неточного изложения требований или их изменения в течение длительного периода создания ИС заказчик получает систему, не удовлетворяющую его потребностям.

**Инкрементная модель** подразумевает разработку информационной системы с линейной последовательностью стадий, но в несколько инкрементов (версий), т. е. с запланированным улучшением продукта.

На рисунке 2 представлен снимок инкрементной модели.



Рисунок 2 – Инкрементная модель

В начале работы над проектом определяются все основные требования к системе, после чего выполняется ее разработка в виде последовательности версий. При этом каждая версия является законченным и работоспособным продуктом. Первая версия реализует часть запланированных возможностей, следующая версия реализует дополнительные возможности и т. д., пока не будет получена полная система.

Данная модель жизненного цикла характерна при разработке сложных и комплексных систем, для которых имеется четкое видение (как со стороны заказчика, так и со стороны разработчика) того, что собой должен представлять конечный результат (информационная система).

Достоинства и недостатки этой модели такие же, как и у предыдущей. Но в отличие от линейной модели заказчик может раньше увидеть результаты. Уже по результатам разработки и внедрения первой версии он может незначительно изменить требования к разработке, отказаться от нее или предложить разработку более совершенного продукта с заключением нового договора.

**Спиральная модель** подразумевает разработку в виде последовательности версий, но в начале проекта определены не все требования. Требования уточняются в результате разработки версий.

На рисунке 3 представлен снимок спиральной модели.

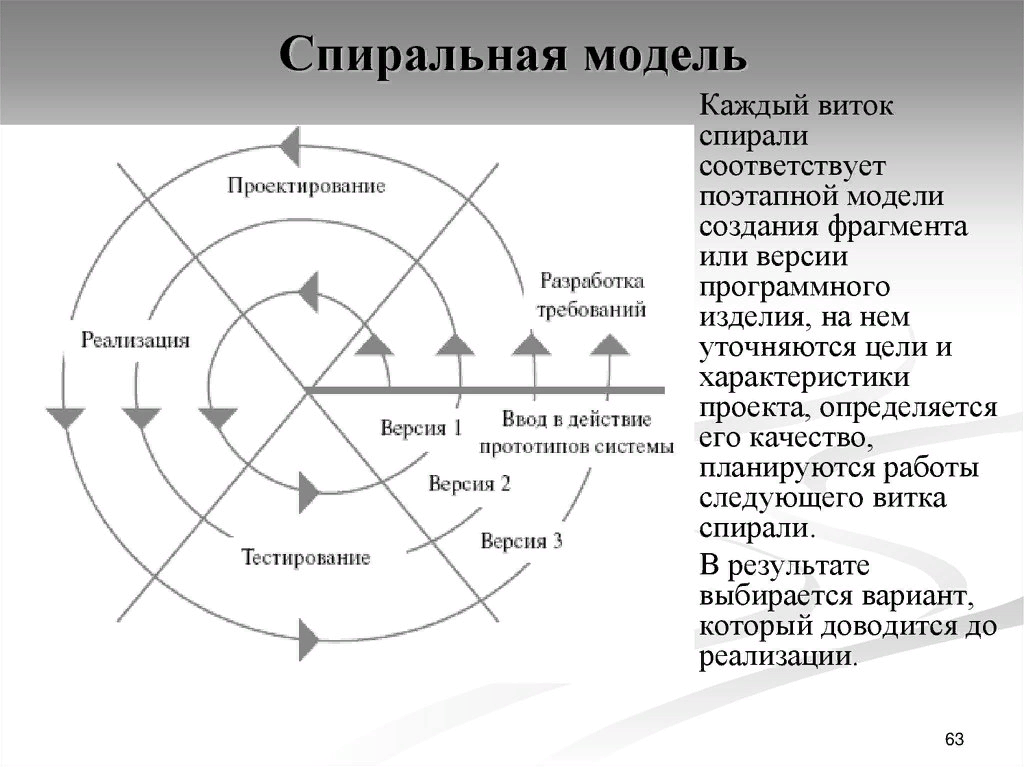


Рисунок 3 – Спиральная модель

Данная модель жизненного цикла характерна при разработке новаторских (нетиповых) систем. В начале работы над проектом у заказчика и разработчика нет четкого видения итогового продукта (требования не могут быть четко определены) или стопроцентной уверенности в успешной реализации проекта (риски очень велики). В связи с этим принимается решение разработки системы по частям с возможностью изменения требований или отказа от ее дальнейшего развития. Развитие проекта может быть завершено не только после стадии внедрения, но и после стадии анализа риска.

Достоинства модели:

* позволяет быстрее показать пользователям системы работоспособный продукт, тем самым, активизируя процесс уточнения и дополнения требований;
* допускает изменение требований при разработке информационной системы, что характерно для большинства разработок, в том числе и типовых;
* обеспечивает большую гибкость в управлении проектом;
* позволяет получить более надежную и устойчивую систему. По мере развития системы ошибки и слабые места обнаруживаются и исправляются на каждой итерации;
* позволяет совершенствовать процесс разработки—анализ, проводимый в каждой итерации, позволяет проводить оценку того, что должно быть изменено в организации разработки, и улучшить ее на следующей итерации;
* уменьшаются риски заказчика. Заказчик может с минимальными для себя финансовыми потерями завершить развитие неперспективного проекта.

Недостатки модели:

* увеличивается неопределенность у разработчика в перспективах развития проекта. Этот недостаток вытекает из предыдущего достоинства модели;
* затруднены операции временного и ресурсного планирования всего проекта в целом. Для решения этой проблемы необходимо ввести временные ограничения на каждую из стадий жизненного цикла. Переход осуществляется в соответствии с планом, даже если не вся запланированная работа выполнена. План составляется на основе статистических данных, полученных в предыдущих проектах и личного опыта разработчиков

## 1.3 Автоматизированные информационные системы (АИС) и информационное моделировании.

**Автоматизированная информационная система**— совокупность программно-аппаратных средств, предназначенных для автоматизации деятельности, связанной с хранением, передачей и обработкой информации.

В АИС входят следующие основные компоненты:

* аппаратные средства вычислительной техники;
* аппаратные средства телекоммуникации (связи);
* программные средства реализации функций АИС;
* информационные базы данных (БД);
* документация, регламентирующая функции и применение компонент АИС;
* специалисты, обслуживающие и использующие программно-технические средства.

**Предметная область АИС** — это совокупность объектов реального или предполагаемого мира, рассматриваемых в пределах данного контекста, который понимается как отдельное рассуждение, фрагмент научной теории или теории в целом и ограничивается рамками данного контекста.

Классификация АИС(по признаку применения):

* **автоматизированная система управления (АСУ)**—организационно-техническая система, созданная с применением автоматизированных информационных технологий для повышения эффективности процессов управления различными объектами;
* **автоматизированная система научных исследований (АСНИ)**— АИС, предназначенная для информационно-аналитического обеспечения научно-исследовательских работ; экспертная система — АИС, которая использует экспертные знания для обеспечения высокоэффективного решения задач в узкой предметной области;
* **автоматизированная система контроля измерений**— АИС, предназначенная для сбора, анализа и хранения показаний контрольно-измерительных приборов;
* **система автоматизированного проектирования (САПР)**—организационно - техническая система, состоящая из программно-технического комплекса автоматизации проектирования, пользователями которого являются сотрудники подразделений проектной организации;
* **автоматизированная система обучения**— АИС, которая включает студентов, преподавателей, комплекс учебно-методических и дидактических материалов, автоматизированную систему обработки данных и предназначена для обеспечения процесса обучения с целью повышения его эффективности;
* **автоматизированная справочная система**— справочное руководство, содержание которого создается, хранится и доводится до пользователя с использованием автоматизированных информационных технологий;
* **автоматизированная библиотечная система**— АИС, обеспечивающая доступ к данным библиотечных каталогов и фондов, а также сбор, обработку и хранение соответствующей информации;
* **автоматизированная система перевода**— АИС, предназначенная для перевода текстов с одного языка на другой; составной частью такой системы является автоматизированный словарь; автоматизированная информационная юридическая система — АИС в предметной области юриспруденции;
* **автоматизированные системы военного назначения**— АИС, предназначенные для управления боевыми действиями, военными объектами, системами ПВО и т.д.

Основные виды обеспечения АИС:

1. Информационное обеспечение АИС.

Виды информационного обеспечения АИС:

* **База данных** - поименованная целостная, единая система данных, организованная по определенным правилам, которые предусматривают общие принципы описания, хранения и обработки данных.
* **База знаний** - формализованная система сведений о некоторой предметной области, содержащая данные о свойствах объектов, закономерностях процессов и явлений и правила использования в задаваемых ситуациях этих данных для принятия новых решений.

2. Средства обработки данных.

3. Интерфейсы пользователя.

**Интерфейс пользователя** — это программно-технические средства, которые обеспечивают взаимодействие пользователя с АИС.

В АИС рассматриваются и другие интерфейсы:

* межкомпонентные интерфейсы;
* интерфейсы между различными АИС;
* интерфейсы в телекоммуникационных сетях.

**Модель данных** — это логическое представление данных и совокупность операций над ними.

Существует несколько аспектов моделирования в обработке данных:

* информационное моделирование:
* концептуальное моделирование (моделирование семантики предметной области);
* логическое моделирование данных;
* физическое моделирование:
* создание моделей доступа к данным;
* оптимизация физической организации данных в аппаратной среде.

Объектами информационной модели являются сущности реального мира из предметной области. Свойства объектов (сущностей) называют атрибутами.

В рамках информационного моделирования существует несколько точек зрения (схем) на абстрагирование данных.

С точки зрения пользователя (называемой внешней схемой), определение данных представляется в контексте языка предметной области. Структура данных и содержание меняется в зависимости от сферы деятельности и особенностей конкретного пользователя. С точки зрения компьютера (называемой внутренней схемой), данные определяются в терминах файловых структур для хранения и поиска. Структура данных в этом случае зависит от конкретной компьютерной технологии и от требований эффективности обработки данных.

Процесс информационного моделирования состоит из нескольких этапов. Обобщенное неформальное описание создаваемой базы данных, выполненное с использованием естественного языка, математических формул, таблиц, графиков и других средств, понятных всем людям, работающих над проектированием базы данных, называют **инфологической моделью данных**.

На рисунке 4 представлен снимок концепции трех схем.

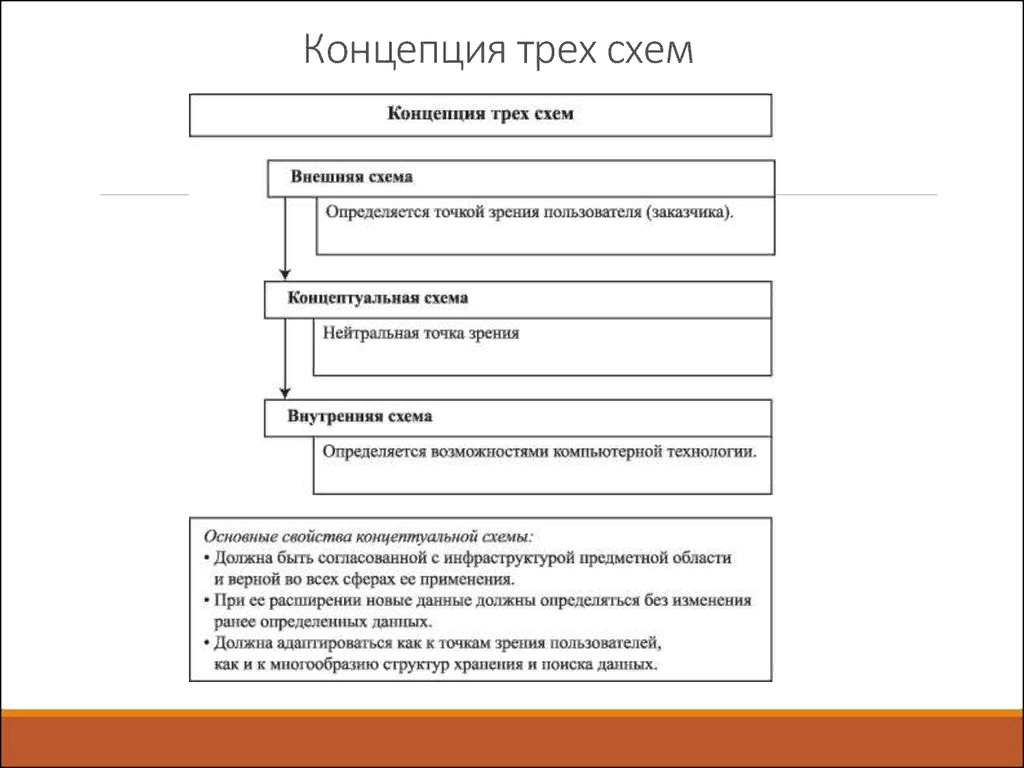


Рисунок 4 – Концепция трех схем

Такая модель полностью независима от физических параметров среды хранения данных. Поэтому инфологическая модель не должна изменяться до тех пор, пока какие-то изменения в реальном мире не потребуют изменения в ней некоторого определения, чтобы эта модель продолжала отражать предметную область.

Остальные модели, показанные на рисунке ниже, являютсякомпьютеро-ориентированными**.** С их помощью СУБД дает возможность программам и пользователям осуществлять доступ к хранимым данным лишь по их именам, не заботясь о физическом расположении этих данных. Нужные данные отыскиваются СУБД на внешних запоминающих устройствах по физической модели данных. Так как указанный доступ осуществляется с помощью конкретной СУБД, то модели должны быть описаны на языке описания данных этой СУБД. Такое описание, создаваемое АБД по инфологической модели данных, называют **даталогической моделью данных.**

Трехуровневая архитектура (инфологический, даталогический и физический уровни) позволяет обеспечить независимость хранимых данных использующих их программ.

**Инфологическая модель данных «Сущность-связь»**

Цель инфологического моделирования — обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком (последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка). Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты). Базовыми элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

На рисунке 5 представлен пример инфологической модели «Сущность-связь».



Рисунок 5 – Пример инфологической модели «Сущность-связь»

**Даталогическая модель данных**

Типы **даталогических моделей** — это есть не что иное, как модели представления данных, т.е. даталогическая модель данных может быть реляционной, иерархической или сетевой.

**Иерархическая модель** — это самая первая модель представления данных, в которой все записи базы данных представлены в виде дерева.

На рисунке 6 представлен пример иерархической модели данных.



Рисунок 6 – Пример иерархической модели данных

**Сетевая модель**—это база данных, в которой одна запись может участвовать в нескольких отношениях предок-потомок. Т.е. фактически, база данных представляет собой не дерево, а граф.

На рисунке 7 представлен пример сетевой модели данных.



Рисунок 7 – Пример сетевой модели данных

В реляционной модели данных вся информации представляется в виде таблиц, и любые операции над данными — это операции над таблицами.

**Диаграмма потоков данных**

**Диаграммы потоков данных (Data Flow Diagramming)** являются основным средством моделирования функциональных требований к проектируемой системе.

На рисунке 8 представлен пример диаграммы потоков данных.

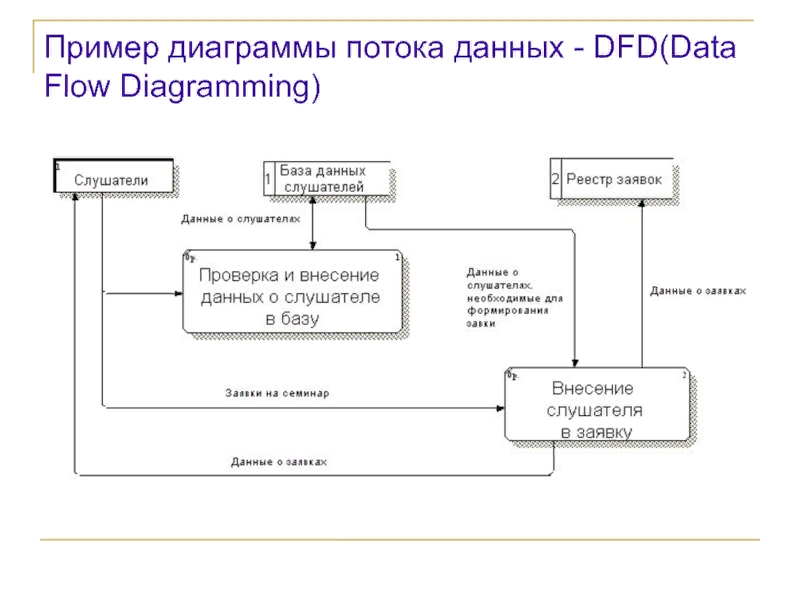


Рисунок 8 – Пример диаграммы потоков данных

**Вывод по первой главе**. Был проведён анализ всего теоретического материала по предложенной теме - “ Моделирование потоков данных в информационных системах ”.

# ГЛАВА 2. ПРАКТИЧЕСКАЯ ЧАСТЬ

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

Чтобы запустить программу на выполнение, необходимо перейти в папку с проектом «WindowsFormsApp4», и запустить файл WindowsFormsApp4.sln, далее откроется приложение Microsoft Visual Studio 2022 с разработкой электронного конспекта и тестовой программой.

## 2.1. Разработка электронного конспекта

Для того, чтобы разработать электронный конспект, понадобилось создать элемент управления, в который был занесен теоретический материал по данной теме.

На форме представлены два элемента button и richTextBox. Это продемонстрировано на рисунке 9.

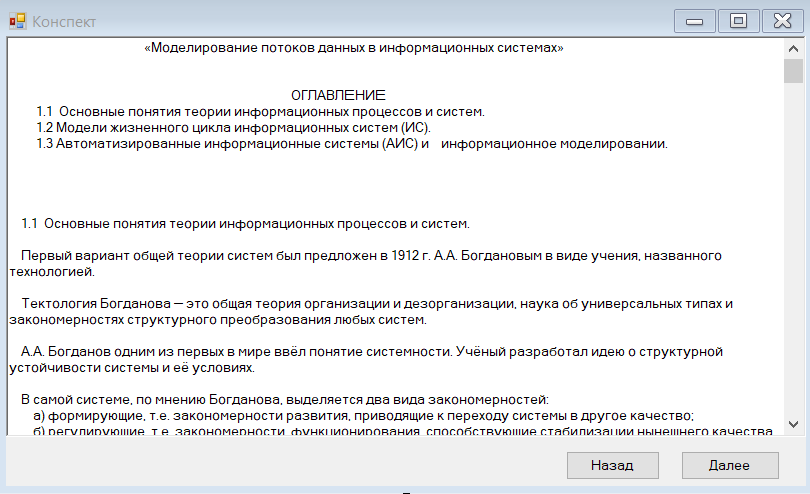


Рисунок 9 – Разработка электронного пособия

С помощью элемента управления richTextBox при переходе на данную форму пользователь видит теоретическую часть. После ознакомления с теоретической частью пользователь может приступить к выполнению теста с помощью кнопки «Далее».

После ознакомления с теоретической частью пользователь может приступить к выполнению теста с помощью кнопки «Далее» или вернуться на титульный лист с помощью кнопки «Назад».

## 2.2. Разработка тестовой программы

Для доступа к каждому следующему вопросу пользователю необходимо ответить на текущий вопрос, следуя указаниям в задании, и нажать кнопку «Ответ». В противном случае программа выдает сообщение: «Вы не ответили на вопрос». Кнопка «Пропустить вопрос» отсутствует в данной программе.

**Перейдем к описанию** работы элемента управления CheckBox. Для того, чтобы пользователь смог ответить на вопрос, необходимо поставить галочку напротив правильных вариантов ответа. Этот вопрос показан на рисунке 10.

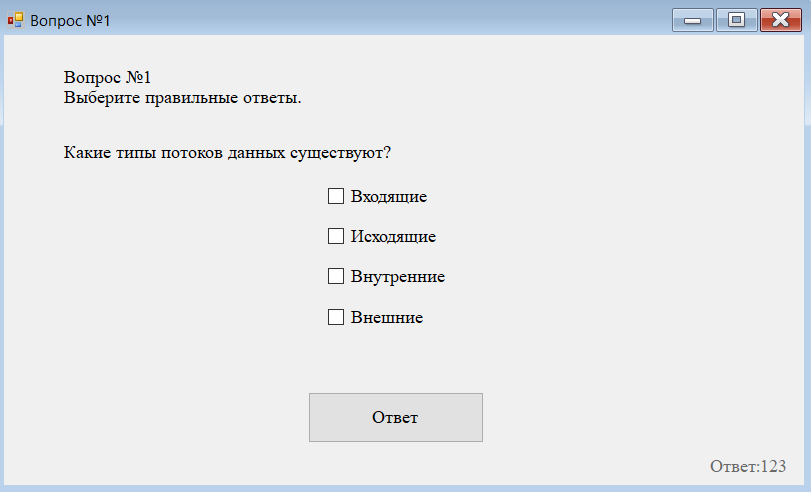


Рисунок 10 - Элемент управления CheckBox

**Код библиотеки классов**

public static int Vopros1(CheckBox CB1, CheckBox CB2, CheckBox CB3, CheckBox CB4)

{

if ((!CB1.Checked) && (!CB2.Checked) && (!CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы не выбрали ни один вариант ответа", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else if ((CB1.Checked) && (CB2.Checked) && (CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы ответили верно", "Верный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[1] = "Верно";

return 1;

}

else

{

MessageBox.Show("Ваш ответ неверный", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[0] = "Неверно";

return 0;

}

}

**Код Windows Forms**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros1(checkBox1, checkBox2, checkBox3, checkBox4);

if (!(rez == -1))

{

Vopros2 f = new Vopros2();

this.Hide();

f.Show();

}

}

Следующим использованным элементом управления является TexBox. Для того, чтобы пользователь смог ответить на вопрос, необходимо в текстовое поле ввести прописные буквы или цифры. Этот вопрос показан на рисунке 11.

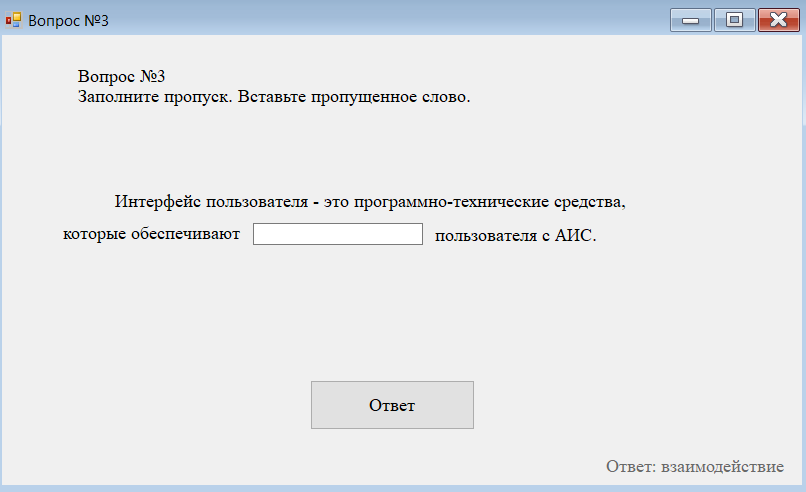


Рисунок 11 – Элемент управления TextBox

**Код библиотеки классов**

public static int Vopros3(string right\_ans, string ans)

{

if (ans == "")

{

MessageBox.Show("Вы не ввели ответ", "Отсутствие ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else

if (ans == right\_ans)

{

MessageBox.Show("Вы ответили верно", "Верный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[0] = "Верно";

return 1;

}

else

{

MessageBox.Show("Ваш ответ неверный", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[0] = "Неверно";

return 0;

}

}

**Код Windows Forms**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros3("взаимодействие", textBox1.Text);

if (!(rez == -1))

{

Vopros4 f = new Vopros4();

this.Hide();

f.Show();

}

}

Далее был использован элемент управления RadioButton. Для того, чтобы пользователь ответил на вопрос, следует выбрать один вариант ответа, нажав на кружок. Форма с описанным элементом управления показана на рисунке 12.

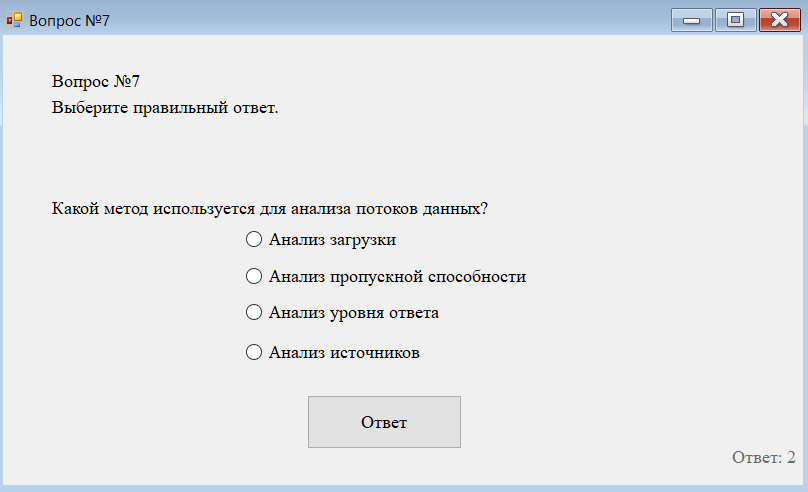


Рисунок 12 – Элемент управления RadioButton

**Код библиотеки классов**

public static int Vopros7(RadioButton CB1, RadioButton CB2, RadioButton CB3, RadioButton CB4)

{

if ((!CB1.Checked) && (!CB2.Checked) && (!CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы не выбрали ни один вариант ответа", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else if ((!CB1.Checked) && (CB2.Checked) && (!CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы ответили верно", "Верный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[1] = "Верно";

return 1;

}

else

{

MessageBox.Show("Ваш ответ неверный", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[0] = "Неверно";

return 0;

}

}

**Код Windows Forms**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros7(radioButton1, radioButton2, radioButton3, radioButton4);

if (!(rez == -1))

{

Vopros8 f = new Vopros8();

this.Hide();

f.Show();

}

}

Следующий элемент управления – ListBox. Для того, чтобы ответить на вопрос, необходимо из окна сверху выбрать определение, то есть один из вариантов ответа и, щёлкнув по соответствующей кнопке “Выбрать” рядом с желаемым положением. Пользователь может выбрать любой из элементов в окне сверху. Использование элемента управления ListBox показано на рисунке 13.

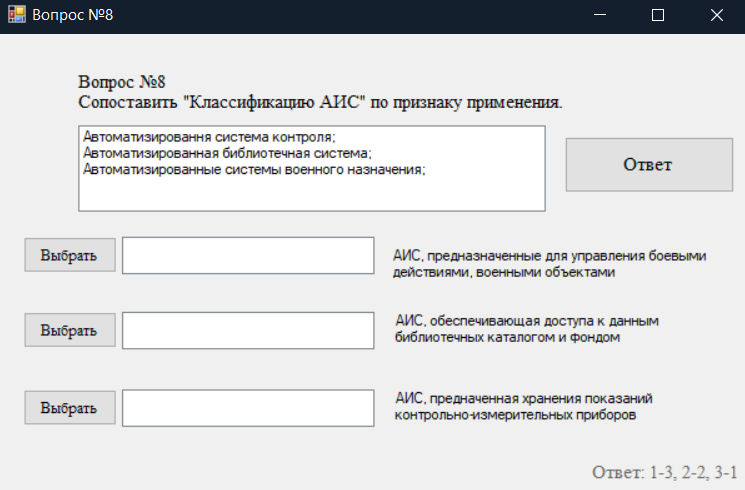


Рисунок 13 – Элемент управления ListBox

**Код Windows Forms**

private void Vopros8\_Load(object sender, EventArgs e)

{

string[] itemlist1 = new string[3];

itemlist1[0] = "Автоматизировання система контроля; ";

itemlist1[1] = "Автоматизированная библиотечная система; ";

itemlist1[2] = "Автоматизированные системы военного назначения; ";

for (int j = 0; j <= 2; j++)

listBox1.Items.Add(itemlist1[j]);

}

private void button2\_Click\_1(object sender, EventArgs e)

{

if (listBox1.SelectedIndex == 0)

{

listBox2.Items.Add(listBox1.Text);

listBox2.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(0);

}

if (listBox1.SelectedIndex == 1)

{

listBox2.Items.Add(listBox1.Text);

listBox2.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(1);

}

if (listBox1.SelectedIndex == 2)

{

listBox2.Items.Add(listBox1.Text);

listBox2.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(2);

}

}

private void button3\_Click\_1(object sender, EventArgs e)

{

if (listBox1.SelectedIndex == 0)

{

listBox3.Items.Add(listBox1.Text);

listBox3.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(0);

}

if (listBox1.SelectedIndex == 1)

{

listBox3.Items.Add(listBox1.Text);

listBox3.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(1);

}

if (listBox1.SelectedIndex == 2)

{

listBox3.Items.Add(listBox1.Text);

listBox3.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(2);

}

}

private void button4\_Click\_1(object sender, EventArgs e)

{

if (listBox1.SelectedIndex == 0)

{

listBox4.Items.Add(listBox1.Text);

listBox4.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(0);

}

if (listBox1.SelectedIndex == 1)

{

listBox4.Items.Add(listBox1.Text);

listBox4.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(1);

}

if (listBox1.SelectedIndex == 2)

{

listBox4.Items.Add(listBox1.Text);

listBox4.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(2);

}

}

private void button1\_Click\_1(object sender, EventArgs e)

{

if ((listBox2.Text == "") || (listBox3.Text == "") || (listBox4.Text == ""))

{

MessageBox.Show("Вы не ответили или ответили не до конца", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else if

((listBox2.Items[0] == "Автоматизированные системы военного назначения; ") && (listBox3.Items[0] == "Автоматизированная библиотечная система; ")

&& (listBox4.Items[0] == "Автоматизировання система контроля; "))

{

MessageBox.Show("Вы ответили правильно!", "Верный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

Class1.n += 1;

Class1.mass[7] = "Верно";

}

else

{

MessageBox.Show("Вы ошиблись", "Неверный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

Class1.mass[7] = "Неверно";

}

Vopros9 f = new Vopros9();

this.Hide();

f.Show();

}

}

}

Перейдем к элементу управления TrackBar. Для того, чтобы пользователь смог ответить на вопрос, нужно передвинуть ползунок вправо. Выбранное положение подписано ниже элемента управления. Это показано на рисунке 14.

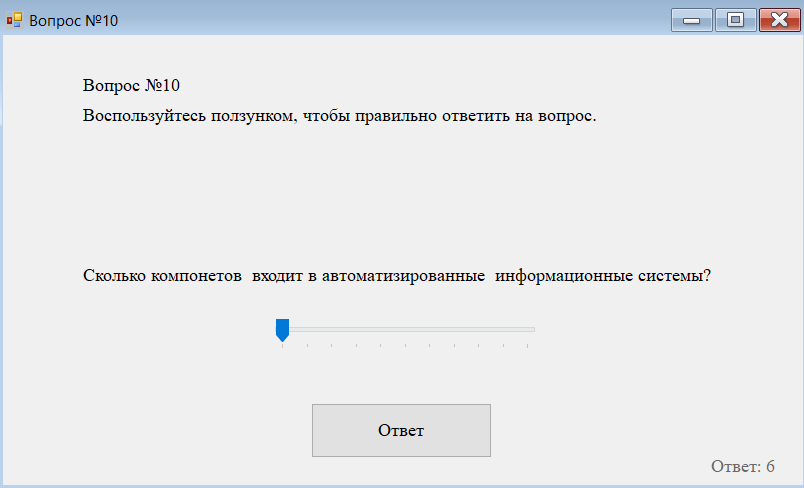


Рисунок 14 – Элемент управления TrackBar

**Код библиотеки классов**

public static int Vopros10(TrackBar D)

{

if (D.Value == 6)

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[9] = "Верно";

return 1;

}

else

{

MessageBox.Show("Вы ответили неправильно", "Неправильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[9] = "Неверно";

return 0;

}

}

**Код Windows Forms**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros10(trackBar1);

{

Vopros11 f = new Vopros11();

this.Hide();

f.Show();

}

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

label1.Text = String.Format("{0}", trackBar1.Value);

}

Перейдем к описанию работы элемента управления ComboBox. Для того, чтобы пользователь смог ответить на вопрос, нужно нажать на стрелочку в поле ответа и из списка предложенных ответов выбрать нужный вариант. Работа элемента управления показана на рисунке 15.

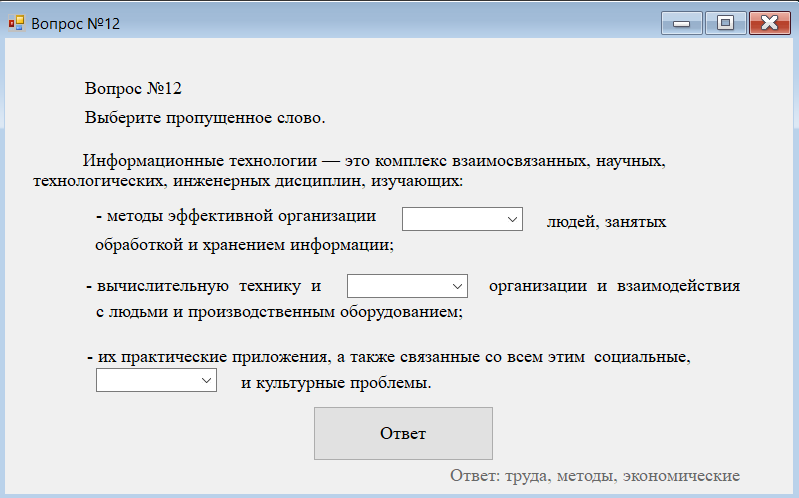


Рисунок 15 – Элемент управления ComboBox

**Код библиотеки классов**

public static int Vopros12(ComboBox B1, ComboBox B2, ComboBox B3)

{

if (B1.SelectedIndex == -1 && B2.SelectedIndex == -1 && B3.SelectedIndex == -1)

{

MessageBox.Show("Вы не выбрали ни один вариант ответа", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else if (B1.SelectedIndex == 1 && B2.SelectedIndex == 2 && B3.SelectedIndex == 0)

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[11] = "Верно";

return 1;

}

else

{

MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[11] = "Неверно";

return 0;

}

}

**Код Windows Forms**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros12(comboBox1, comboBox2, comboBox3);

{

Vopros13 f = new Vopros13();

this.Hide();

f.Show();

}

}

Элемент управления HScrollBar. Для того, чтобы пользователь смог ответить на вопрос, нужно передвинуть полосу в нужное положение со значением. Выбранное положение подписано ниже элемента управления. Это представлено на рисунке 16.

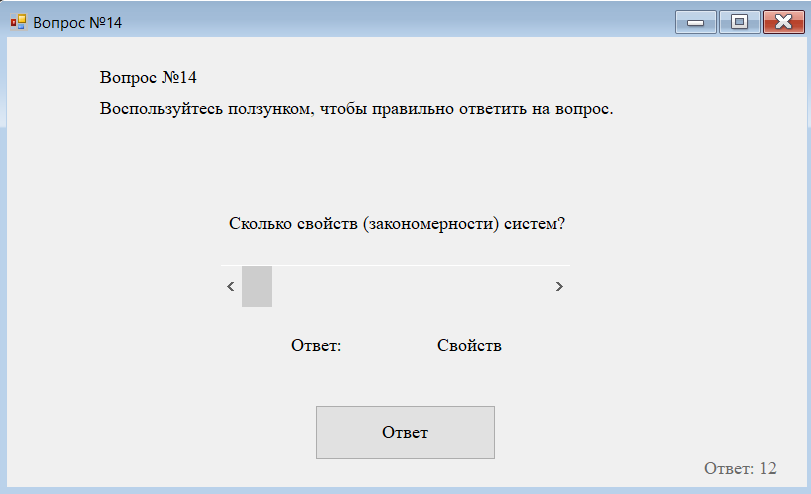


Рисунок 16 – Элемент управления HScrollBar

**Код библиотеки классов**

public static int Vopros14(HScrollBar D)

{

if (D.Value == 12)

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[13] = "Верно";

return 1;

}

else

{

MessageBox.Show("Вы ответили неправильно", "Неправильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[13] = "Неверно";

return 0;

}

}

**Код Windows Forms**

private void button1\_Click(object sender, EventArgs e)

{

{

int rez = Class1.Vopros14(hScrollBar1);

{

Vopros15 f = new Vopros15();

this.Hide();

f.Show();

}

}

}

private void hScrollBar1\_Scroll(object sender, ScrollEventArgs e)

{

label2.Text = String.Format("{0}", hScrollBar1.Value);

}

Последним элементом управления является CheckedListBox. Для того, чтобы пользователь смог ответить на вопрос, нужно кликнуть левой кнопкой мыши и поставить галочку напротив нужных вариантов ответа. Это показано на рисунке 17.

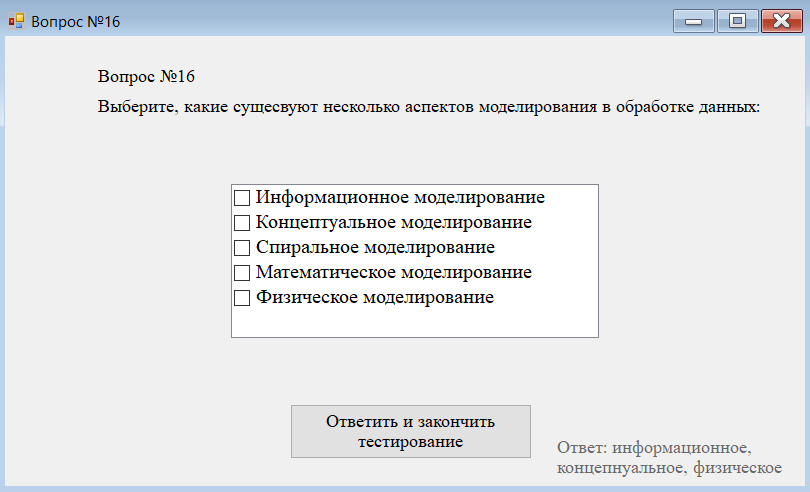


Рисунок 17 – Элемент управления CheckedListBox

**Код библиотеки классов**

public static int Vopros16(CheckedListBox D)

{

if (D.SelectedItem == null)

{

MessageBox.Show("Вы не выбрали ни одного ответа", "Нет ответа", MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else

if ((D.GetItemCheckState(0) == CheckState.Checked) && (D.GetItemCheckState(1) == CheckState.Checked) && (D.GetItemCheckState(4) == CheckState.Checked))

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[15] = "Верно";

return 1;

}

else

{

MessageBox.Show("Вы ответили неправильно", "Неправильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[15] = "Неверно";

return 0;

}

}

**Код Windows Forms**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros16(checkedListBox1);

{

Diagramma f = new Diagramma();

this.Hide();

f.Show();

}

}

### На данном этапе перейдем к описанию итоговой формы, которая представлена на рисунках 18-19.

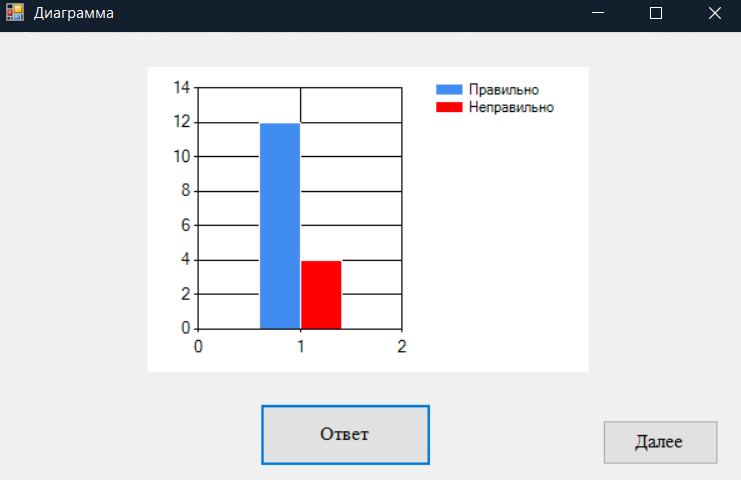


Рисунок 18 – Итоговая форма №1

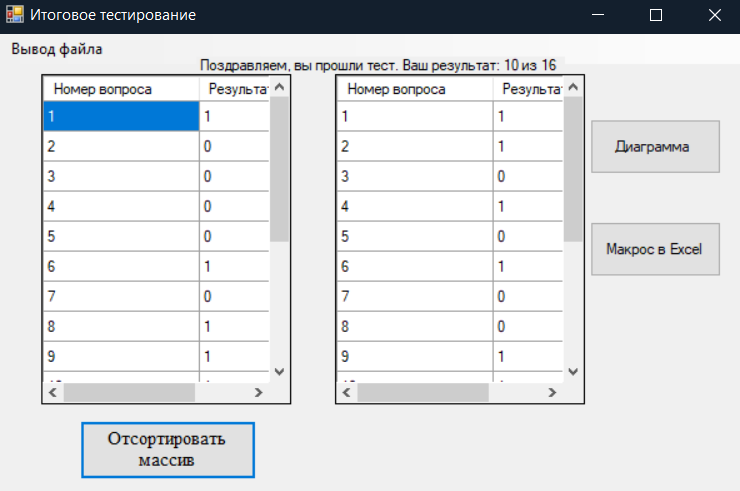


Рисунок 19 – Итоговая форма №2

В крайней форме выводится массив с результатами теста в элемент dataGridView, с помощью которого можно узнать в каких заданиях была допущена ошибка, а также диаграмма соотношения ответов.

## 2.3 Описание дополнительных возможностей

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

Было реализовано:

1. Запись результатов ответов тестовой программы (в формате единицы и нуля) в одномерный массив с выводом в DGV на итоговую форму.

2. Упорядочены результаты теста в виде компонента dataGridView, с использованием метода сортировки бинарными вставками. Создан макрос для форматирования таблиц и установки границ в таблицах, а также создана кнопка на листе Excel, которая выполняет написанный макрос;

3. Построение диаграммы, отображающей результаты тестирования в Microsoft Excel и с помощью элемента управления Chart;

Перейдем к **рассмотрению вышеперечисленных** возможностей.

Прежде всего, рассмотрим элемент управления DataGridView, пользователь на итоговой форме может увидеть в каких заданиях была допущена ошибка. Запись результатов была продемонстрирована на рис. 19.

Программный код для реализации данной задачи приведен ниже:

**Код библиотеки классов**

public static int n = 0;

public static string[] mass = new string[16];

**Код Windows Forms**

private void button1\_Click(object sender, EventArgs e)

{

label1.Text = "Поздравляем, вы прошли тест. Ваш результат: " + Kursovay\_Rabota.Class1.n.ToString() + " из 16 ";

dataGridView1.Visible = true;

button1.Visible = false;

Kursovay\_Rabota.Class1.VivodDGV(dataGridView1);

}

Далее, на форме  упорядоченные результаты теста, представленные в виде одномерного массива в другом компоненте dataGridView, с использованием метода сортировки бинарными вставками, при этом используется алгоритм сортировки. Отсортированный массив записан в табличный процессор Microsoft Excel. Создан макрос для форматирования таблиц и установки границ в таблицах, а также создана кнопка на листе Excel, которая выполняет написанный макрос

**Код библиотеки классов**

public static void Bin(DataGridView t)

{

for (int i = 2; i < mass.Length; i++)

{

int x = mass[i];

int left = 1;

int right = i - 1;

while (left < right)

{

int m = (left + right) / 2;

if (x < mass[m])

right = m - 1;

else

left = m + 1;

}

for (int j = i - 1; j < left - 1; j++)

{

mass[j + 1] = mass[j];

}

mass[left] = x;

}

for (int s = 0; s < mass.Length; s++)

{

t.Rows.Add((s + 1).ToString(), mass[s]);

}

}

public static void Zap\_excel(int[,] mas, int col, int rows)

{

Microsoft.Office.Interop.Excel.Application excelApp = new Microsoft.Office.Interop.Excel.Application();

String st = System.IO.Directory.GetCurrentDirectory() + "\\Macros1.xlsm";

Microsoft.Office.Interop.Excel.Workbook workbook = excelApp.Workbooks.Open(st);

Microsoft.Office.Interop.Excel.Worksheet worksheet;

worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.Worksheets.get\_Item(1);

worksheet.Name = "Массив исходный";

for (int i = 0; i < col; i++)

for (int j = 0; j < rows; j++)

worksheet.Cells[i + 1, j + 1].Value = mas[i, j];

worksheet.Range["A15"].Select();

excelApp.Visible = true;

excelApp.UserControl = true;

}

}

}

**Код Windows Forms**

private void button3\_Click(object sender, EventArgs e)

{

Class1.Zap\_excel(arr2,15,2);

}

private void button4\_Click(object sender, EventArgs e)

{

Class1.Bin(dataGridView2);

}

}

}

Перейдем к описанию заключительной дополнительной возможности построение диаграммы. На итоговой форме тестирования пользователь может увидеть наглядный результат прохождения теста в виде диаграммы в Excel или же с помощью элемента управления Chart. В свойстве Series были переименованы на «Правильно» и «Неправильно» члены диаграммы. Построенная диаграмма была продемонстрирована на рис. 18.

Программный код для реализации данной задачи приведен ниже:

**Код Windows Forms**

private void button1\_Click(object sender, EventArgs e)

{

chart1.Visible = true;

chart1.Series["Правильно"].Points.AddY(Class1.n);

chart1.Series["Неправильно"].Points.AddY(16 - Class1.n);

}

private void button2\_Click(object sender, EventArgs e)

{

Microsoft.Office.Interop.Excel.Application excelApp = new

Microsoft.Office.Interop.Excel.Application();

excelApp.Visible = true;

var t = Type.Missing;

var Book = excelApp.Workbooks.Add(t);

var Lists = Book.Worksheets;

Worksheet List = Lists.Item[1];

List.Cells[1, 1] = "Результат тестирования";

List.Range["A2", t].Value2 = "Вопрос 1";

List.Range["A3", t].Value2 = "Вопрос 2";

List.Range["A4", t].Value2 = "Вопрос 3";

List.Range["A5", t].Value2 = "Вопрос 4";

List.Range["A6", t].Value2 = "Вопрос 5";

List.Range["A7", t].Value2 = "Вопрос 6";

List.Range["A8", t].Value2 = "Вопрос 7";

List.Range["A9", t].Value2 = "Вопрос 8";

List.Range["A10", t].Value2 = "Вопрос 9";

List.Range["A11", t].Value2 = "Вопрос 10";

List.Range["A12", t].Value2 = "Вопрос 11";

List.Range["A13", t].Value2 = "Вопрос 12";

List.Range["A14", t].Value2 = "Вопрос 13";

List.Range["A15", t].Value2 = "Вопрос 14";

List.Range["A16", t].Value2 = "Вопрос 15";

List.Range["A17", t].Value2 = "Вопрос 16";

List.Range["B1", t].Value2 = "Ответ";

List.Range["B2", t].Value2 = Class1.mass[0];

List.Range["B3", t].Value2 = Class1.mass[1];

List.Range["B4", t].Value2 = Class1.mass[2];

List.Range["B5", t].Value2 = Class1.mass[3];

List.Range["B6", t].Value2 = Class1.mass[4];

List.Range["B7", t].Value2 = Class1.mass[5];

List.Range["B8", t].Value2 = Class1.mass[6];

List.Range["B9", t].Value2 = Class1.mass[7];

List.Range["B10", t].Value2 = Class1.mass[8];

List.Range["B11", t].Value2 = Class1.mass[9];

List.Range["B12", t].Value2 = Class1.mass[10];

List.Range["B13", t].Value2 = Class1.mass[11];

List.Range["B14", t].Value2 = Class1.mass[12];

List.Range["B15", t].Value2 = Class1.mass[13];

List.Range["B16", t].Value2 = Class1.mass[14];

List.Range["B17", t].Value2 = Class1.mass[15];

Microsoft.Office.Interop.Excel.Chart Diagr = excelApp.Charts.Add(t, t, t, t);

Diagr.SetSourceData(List.Range["A2", "B17"],

Microsoft.Office.Interop.Excel.XlRowCol.xlColumns);

Diagr.ChartType = XlChartType.xlColumnClustered;

Diagr.HasLegend = false;

Diagr.HasTitle = true;

Diagr.ChartTitle.Caption = "РЕЗУЛЬТАТЫ ТЕСТА";

Axis Goriz\_Os = Diagr.Axes(Microsoft.Office.Interop.Excel.XlAxisType.xlValue, Microsoft.Office.Interop.Excel.XlAxisGroup.xlPrimary);

Goriz\_Os.HasTitle = true;

Goriz\_Os.AxisTitle.Text = "ОТВЕТЫ";

Axis Vertic\_Os = Diagr.Axes(Microsoft.Office.Interop.Excel.XlAxisType.xlValue, Microsoft.Office.Interop.Excel.XlAxisGroup.xlPrimary);

Vertic\_Os.HasTitle = true;

Vertic\_Os.AxisTitle.Text = "ВОПРОСЫ";

excelApp.ActiveChart.Export(@"C:\Users\evgen\excelfile\Excel.jpg", t, t);

}

По итогам работы над второй главой можно отметить, что был разработан электронный конспект и создана программа для тестирования пользователя по теме «Моделирование потоков данных в информационных системах». При этом были реализованы следующие задачи:

1. Разработан электронный конспект по теме «Моделирование потоков данных в информационных системах»;

2. Разработана тестовая программа, состоящая из 16 вопросов и дополнительного функционала

3. Записаны результаты ответа на каждый вопрос в элемент управления dataGridView;

4. Разработана возможность записи результата ответов на тестовые вопросы в формате единицы и нуля в Microsoft Excel;

5. Создана диаграмма, отображающая результаты теста в Microsoft Excel и с помощью элемента управления Chart;

6. Упорядочены результаты теста в виде компонента dataGridView, с использованием метода сортировки бинарными вставками. Создан макрос для форматирования таблиц и установки границ в таблицах, а также создана кнопка на листе Excel, которая выполняет написанный макрос.

**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:

1. Проанализирован и структурирован теоретический материал по теме исследования;

2. Разработано электронный конспект и программа, проверяющая усвоение знаний по данной теме;

3. Реализована запись результатов ответов в одномерный массив с выводом на элемент управления dataGridView;

4. Разработана возможность записи результата ответов на тестовые вопросы в формате единицы и нуля в Microsoft Excel;

5. Реализовано построение диаграммы, отображающей результаты тестирования в Microsoft Excel и с помощью элемента управления Chart;

6. Упорядочены результаты теста в виде компонента dataGridView, с использованием метода сортировки бинарными вставками. Создан макрос для форматирования таблиц и установки границ в таблицах, а также создана кнопка на листе Excel, которая выполняет написанный макрос.

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

## Список используемых источников

1. Гуриков С. Р. Введение в программирование на языке Visual C#: учебное пособие / С.Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2020. — 447 с.
2. ГОСТ 7.32-2017, Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Условные обозначения и правила выполнения / Москва.; Стандартинформ 2017.; – 26 с.

## Приложение А

Задание 1. Выберите правильные ответы.

Какие типы потоков данных существуют?

А) Входящие

Б) Исходящие

В) Внутренние

Г) Внешние

Ответ: 123

Задание 2. Выберите правильные ответы.

Какими определениями обладает термин "Система" по мнению австралийского биолога и философа А.А. Богданова?

Ответ: 134

Задание 3. Заполните пропуск. Вставьте пропущенное слово.

Интерфейс пользователя - это программно-технические средства, которые обеспечивают … пользователя с АИС.

Ответ: взаимодействие

Задание 4. Заполните пропуск. Вставьте пропущенное слово.

Модель данных — это логическое представление данных и совокупность … над ними.

Ответ: операций

Задание 5. Выберите правильный ответ.

Какого аспекта в моделировании обработки данных не существует?

А) Информационное моделирование

Б) Логическое моделирование данных

В) Телекоммуникационное моделирование

Г) Создание моделей доступа к данным

Ответ: 3

Задание 6. Выберите правильный ответ.

Назовите модель жизненного цикла информационной системы.

А) Каскадная модель

Б) Графическая модель

В) Чертёжная модель

Г) Образная модель

Ответ: 1

Задание 7. Выберите правильный ответ.

Какой метод используется для анализа потоков данных?

А) Анализ загрузки

Б) Анализ пропускной способности

В) Анализ уровня ответа

Г) Анализ источников

Ответ: 2

Задание 8. Сопоставить "Классификацию АИС" по признаку применения.

1) Автоматизированная система контроля

2) Автоматизированная библиотечная система

3) Автоматизированные системы военного назначения

Ответ: 321

Задание 9. Сопоставить "Cтадии жизненного цикла информационной системы" с их определениями.

1) Системный анализ

2) Проектирование

3) Внедрение и сопровождение

Ответ: 213

Задание 10. Воспользуйтесь ползунком, чтобы правильно ответить на вопрос.

Сколько компонентов входит в автоматизированные информационные системы?

Ответ: 6

Задание 11. Воспользуйтесь ползунком, чтобы правильно ответить на вопрос.

Сколько интерфейсов рассматривается в автоматизированных информационных системах?

Ответ: 3

Задание 12. Выберите правильное слово.

Информационные технологии — это комплекс взаимосвязанных, научных, технологических, инженерных дисциплин, изучающих:

1) - методы эффективной организации … людей, занятых обработкой и хранением информации

2) - вычислительную технику и … организации и взаимодействия с людьми и производственным оборудованием

3) - их практические приложения, а также связанные со всем этим социальные, … и культурные проблемы

Ответ: труда, методы, экономические

Задание 13. Выберите правильное слово.

А) … модель подразумевает линейную последовательность выполнения стадий создания информационной системы.

Б) … модель подразумевает разработку информационной системы с линейной последовательностью стадий, но в несколько инкрементов (версий), т. е. с запланированным улучшением продукта

В) … модель подразумевает разработку в виде последовательности версий, но в начале проекта определены не все требования.

Ответ: каскадная, инкрементная, спиральная

Задание 14. Воспользуйтесь ползунком, чтобы правильно ответить на вопрос.

Сколько свойств (закономерности) систем?

Ответ: 12

Задание 15. Воспользуйтесь ползунком, чтобы правильно ответить на вопрос.

Сколько блоков внутри инкрементной модели?

Ответ: 16

Задание 16. Выберите правильные ответы.

Выберите, какие существуют несколько аспектов моделирования в обработке данных:

А) Информационное моделирование

Б) Концептуальное моделирование

В) Спиральное моделирование

Г) Математическое моделирование

Д) Физическое моделирование

Ответ: Информационное моделирование, концептуальное моделирование, физическое моделирование

## Приложение B

**Код DLL-библиотеки**

using System;

using System.Windows.Forms;

using Axis = Microsoft.Office.Interop.Excel.Axis;

using CheckBox = System.Windows.Forms.CheckBox;

using Microsoft.Office.Interop.Excel;

using System.IO;

using Excel = Microsoft.Office.Interop.Excel;

namespace Kursovay\_Rabota

{

public class Class1

{

**// создание массивов для вывода ответов и методов вывода**

public static int n = 0;

public static int[] mass = new int[16];

public static int[] rezmas;

public static void VivodDGV(DataGridView DGV)

{

for (int i = 0; i < 16; i++)

{

DGV.Rows.Add((i + 1).ToString(), mass[i]);

}

}

public static int Vopros1(CheckBox CB1, CheckBox CB2, CheckBox CB3, CheckBox CB4) **// Метод для вопросов CheckBox, взят из ЭУ**

{

if ((!CB1.Checked) && (!CB2.Checked) && (!CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы не выбрали ни один вариант ответа", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else if ((CB1.Checked) && (CB2.Checked) && (CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[0] = 1;

return 1;

}

else

{

MessageBox.Show("Ваш ответ неверный", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[0] = 0;

return 0;

}

}

public static int Vopros2(CheckBox CB1, CheckBox CB2, CheckBox CB3, CheckBox CB4) **// Метод для вопросов CheckBox, взят из ЭУ**

{

if ((!CB1.Checked) && (!CB2.Checked) && (!CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы не выбрали ни один вариант ответа", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else if ((CB1.Checked) && (!CB2.Checked) && (CB3.Checked) && (CB4.Checked))

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[1] = 1;

return 1;

}

else

{

MessageBox.Show("Ваш ответ неверный", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[1] = 0;

return 0;

}

}

public static int Vopros3(string right\_ans, string ans) **// Метод для вопросов TextBox, взят из ЭУ**

{

if (ans == "")

{

MessageBox.Show("Вы не ввели ответ", "Отсутствие ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else

if (ans == right\_ans)

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[2] = 1;

return 1;

}

else

{

MessageBox.Show("Ваш ответ неверный", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[2] = 0;

return 0;

}

}

public static int Vopros4(string right\_ans, string ans) **// Метод для вопросов TextBox, взят из ЭУ**

{

if (ans == "")

{

MessageBox.Show("Вы не ввели ответ", "Отсутствие ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else

if (ans == right\_ans)

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[3] = 1;

return 1;

}

else

{

MessageBox.Show("Ваш ответ неверный", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[3] = 0;

return 0;

}

}

public static int Vopros5(RadioButton CB1, RadioButton CB2, RadioButton CB3, RadioButton CB4) **// Метод для вопросов RadioButton, взят из ЭУ**

{

if ((!CB1.Checked) && (!CB2.Checked) && (!CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы не выбрали ни один вариант ответа", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else if ((!CB1.Checked) && (!CB2.Checked) && (CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[4] = 1;

return 1;

}

else

{

MessageBox.Show("Ваш ответ неверный", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[4] = 0;

return 0;

}

}

public static int Vopros6(RadioButton CB1, RadioButton CB2, RadioButton CB3, RadioButton CB4) **// Метод для вопросов RadioButton, взят из ЭУ**

{

if ((!CB1.Checked) && (!CB2.Checked) && (!CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы не выбрали ни один вариант ответа", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else if ((CB1.Checked) && (!CB2.Checked) && (!CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[5] = 1;

return 1;

}

else

{

MessageBox.Show("Ваш ответ неверный", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[5] = 0;

return 0;

}

}

public static int Vopros7(RadioButton CB1, RadioButton CB2, RadioButton CB3, RadioButton CB4) **// Метод для вопросов RadioButton, взят из ЭУ**

{

if ((!CB1.Checked) && (!CB2.Checked) && (!CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы не выбрали ни один вариант ответа", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else if ((!CB1.Checked) && (CB2.Checked) && (!CB3.Checked) && (!CB4.Checked))

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[6] = 1;

return 1;

}

else

{

MessageBox.Show("Ваш ответ неверный", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[6] = 0;

return 0;

}

}

public static int Vopros10(TrackBar D) **// Метод для вопросов TrackBar, взят из ЭУ**

{

if (D.Value == 6)

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[9] = 1;

return 1;

}

else

{

MessageBox.Show("Вы ответили неправильно", "Неправильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[9] = 0;

return 0;

}

}

public static int Vopros11(TrackBar D) **// Метод для вопросов TrackBar, взят из ЭУ**

{

if (D.Value == 3)

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[10] = 1;

return 1;

}

else

{

MessageBox.Show("Вы ответили неправильно", "Неправильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[10] = 0;

return 0;

}

}

public static int Vopros12(ComboBox B1, ComboBox B2, ComboBox B3) **// Метод для вопросов ComboBox, взят из ЭУ**

{

if (B1.SelectedIndex == -1 && B2.SelectedIndex == -1 && B3.SelectedIndex == -1)

{

MessageBox.Show("Вы не выбрали ни один вариант ответа", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else if (B1.SelectedIndex == 1 && B2.SelectedIndex == 2 && B3.SelectedIndex == 0)

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[11] = 1;

return 1;

}

else

{

MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[11] = 0;

return 0;

}

}

public static int Vopros13(ComboBox B1, ComboBox B2, ComboBox B3) **// Метод для вопросов ComboBox, взят из ЭУ**

{

if (B1.SelectedIndex == -1 && B2.SelectedIndex == -1 && B3.SelectedIndex == -1)

{

MessageBox.Show("Вы не выбрали ни один вариант ответа", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else if (B1.SelectedIndex == 1 && B2.SelectedIndex == 0 && B3.SelectedIndex == 2)

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[12] = 1;

return 1;

}

else

{

MessageBox.Show("Вы ответили неправильно", "Неправильный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[12] = 0;

return 0;

}

}

public static int Vopros14(HScrollBar D) **// Метод для вопросов HScrollBar, взят из ЭУ**

{

if (D.Value == 12)

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[13] = 1;

return 1;

}

else

{

MessageBox.Show("Вы ответили неправильно", "Неправильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[13] = 0;

return 0;

}

}

public static int Vopros15(HScrollBar D) **// Метод для вопросов HScrollBar, взят из ЭУ**

{

if (D.Value == 16)

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[14] = 1;

return 1;

}

else

{

MessageBox.Show("Вы ответили неправильно", "Неправильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[14] = 0;

return 0;

}

}

public static int Vopros16(CheckedListBox D) **// Метод для вопросов CheckedListBox, взят из ЭУ**

{

if (D.SelectedItem == null)

{

MessageBox.Show("Вы не выбрали ни одного ответа", "Нет ответа", MessageBoxButtons.OK, MessageBoxIcon.Information);

return -1;

}

else

if ((D.GetItemCheckState(0) == CheckState.Checked) && (D.GetItemCheckState(1) == CheckState.Checked) && (D.GetItemCheckState(4) == CheckState.Checked))

{

MessageBox.Show("Вы ответили правильно", "Правильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

n += 1;

mass[15] = 1;

return 1;

}

else

{

MessageBox.Show("Вы ответили неправильно", "Неправильный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

mass[15] = 0;

return 0;

}

}

public static void Output\_mas(int[] mas, int lenght, DataGridView grid) **// Метод для вывода одномерного массива**

{

grid.ColumnCount = lenght;

grid.RowCount = 2;

for (int i = 0; i < lenght; i++)

{

grid.Rows[0].Cells[i].Value = "[" + i + "]";

grid.Rows[1].Cells[i].Value = mas[i];

}

}

public static void Bin(DataGridView t) **// Метод для вывода бинарной вставки**

{

for (int i = 2; i < mass.Length; i++)

{

int x = mass[i];

int left = 1;

int right = i - 1;

while (left < right)

{

int m = (left + right) / 2;

if (x < mass[m])

right = m - 1;

else

left = m + 1;

}

for (int j = i - 1; j < left - 1; j++)

{

mass[j + 1] = mass[j];

}

mass[left] = x;

}

for (int s = 0; s < mass.Length; s++)

{

t.Rows.Add((s + 1).ToString(), mass[s]);

}

}

public static void Zap\_excel(int[,] mas, int col, int rows) **// Вывод в Excel с макросом**

{

Microsoft.Office.Interop.Excel.Application excelApp = new Microsoft.Office.Interop.Excel.Application();

String st = System.IO.Directory.GetCurrentDirectory() + "\\Macros1.xlsm";

Microsoft.Office.Interop.Excel.Workbook workbook = excelApp.Workbooks.Open(st);

Microsoft.Office.Interop.Excel.Worksheet worksheet;

worksheet = (Microsoft.Office.Interop.Excel.Worksheet)workbook.Worksheets.get\_Item(1);

worksheet.Name = "Массив исходный";

for (int i = 0; i < col; i++)

for (int j = 0; j < rows; j++)

worksheet.Cells[i + 1, j + 1].Value = mas[i, j];

worksheet.Range["A15"].Select();

excelApp.Visible = true;

excelApp.UserControl = true;

}

}

}

**Код Windows Forms**

**// Код кнопки для вопроса CheckBox**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros1(checkBox1, checkBox2, checkBox3, checkBox4);

if (!(rez == -1))

{

Vopros2 f = new Vopros2();

this.Hide();

f.Show();

}

}

**// Код кнопки для вопроса CheckBox**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros2(checkBox1, checkBox2, checkBox3, checkBox4);

if (!(rez == -1))

{

Vopros3 f = new Vopros3();

this.Hide();

f.Show();

}

}

**// Код кнопки для вопроса TextBox**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros3("взаимодействие", textBox1.Text);

if (!(rez == -1))

{

Vopros4 f = new Vopros4();

this.Hide();

f.Show();

}

}

**// Код кнопки для вопроса TextBox**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros4("операций", textBox1.Text);

if (!(rez == -1))

{

Vopros5 f = new Vopros5();

this.Hide();

f.Show();

}

}

**// Код кнопки для вопроса RadioButton**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros5(radioButton1, radioButton2, radioButton3, radioButton4);

if (!(rez == -1))

{

Vopros6 f = new Vopros6();

this.Hide();

f.Show();

}

}

**// Код кнопки для вопроса RadioButton**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros6(radioButton1, radioButton2, radioButton3, radioButton4);

if (!(rez == -1))

{

Vopros7 f = new Vopros7();

this.Hide();

f.Show();

}

}

**// Код кнопки для вопроса RadioButton**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros7(radioButton1, radioButton2, radioButton3, radioButton4);

if (!(rez == -1))

{

Vopros8 f = new Vopros8();

this.Hide();

f.Show();

}

}

**// Метод для вопросов listBox, взят из ЭУ**

private void Vopros8\_Load(object sender, EventArgs e)

{

string[] itemlist1 = new string[3];

itemlist1[0] = "Автоматизированная система контроля; ";

itemlist1[1] = "Автоматизированная библиотечная система; ";

itemlist1[2] = "Автоматизированные системы военного назначения; ";

for (int j = 0; j <= 2; j++)

listBox1.Items.Add(itemlist1[j]);

}

private void button2\_Click\_1(object sender, EventArgs e)

{

if (listBox1.SelectedIndex == 0)

{

listBox2.Items.Add(listBox1.Text);

listBox2.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(0);

}

if (listBox1.SelectedIndex == 1)

{

listBox2.Items.Add(listBox1.Text);

listBox2.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(1);

}

if (listBox1.SelectedIndex == 2)

{

listBox2.Items.Add(listBox1.Text);

listBox2.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(2);

}

}

private void button3\_Click\_1(object sender, EventArgs e)

{

if (listBox1.SelectedIndex == 0)

{

listBox3.Items.Add(listBox1.Text);

listBox3.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(0);

}

if (listBox1.SelectedIndex == 1)

{

listBox3.Items.Add(listBox1.Text);

listBox3.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(1);

}

if (listBox1.SelectedIndex == 2)

{

listBox3.Items.Add(listBox1.Text);

listBox3.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(2);

}

}

private void button4\_Click\_1(object sender, EventArgs e)

{

if (listBox1.SelectedIndex == 0)

{

listBox4.Items.Add(listBox1.Text);

listBox4.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(0);

}

if (listBox1.SelectedIndex == 1)

{

listBox4.Items.Add(listBox1.Text);

listBox4.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(1);

}

if (listBox1.SelectedIndex == 2)

{

listBox4.Items.Add(listBox1.Text);

listBox4.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(2);

}

}

private void button1\_Click\_1(object sender, EventArgs e) **// Метод для вопросов listBox, взят из ЭУ**

{

if ((listBox2.Text == "") || (listBox3.Text == "") || (listBox4.Text == ""))

{

MessageBox.Show("Вы не ответили или ответили не до конца", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else if

((listBox2.Items[0] == "Автоматизированные системы военного назначения; ") && (listBox3.Items[0] == "Автоматизированная библиотечная система; ")

&& (listBox4.Items[0] == "Автоматизировання система контроля; "))

{

MessageBox.Show("Вы ответили правильно", "Верный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

Class1.n += 1;

Class1.mass[7] = 1;

}

else

{

MessageBox.Show("Вы ошиблись", "Неверный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

Class1.mass[7] =0;

}

Vopros9 f = new Vopros9();

this.Hide();

f.Show();

}

}

private void Vopros9\_Load(object sender, EventArgs e)

{

string[] itemlist1 = new string[3];

itemlist1[0] = "Cистемный анализ; ";

itemlist1[1] = "Проектирование; ";

itemlist1[2] = "Внедрение и сопровождение; ";

for (int j = 0; j <= 2; j++)

listBox1.Items.Add(itemlist1[j]);

}

private void button1\_Click(object sender, EventArgs e) **// Метод для вопросов listBox, взят из ЭУ**

{

if ((listBox2.Text == "") || (listBox3.Text == "") || (listBox4.Text == ""))

{

MessageBox.Show("Вы не ответили или ответили не до конца", "Нет ответа",

MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else if

((listBox2.Items[0] == "Проектирование; ") && (listBox3.Items[0] == "Cистемный анализ; ")

&& (listBox4.Items[0] == "Внедрение и сопровождение; "))

{

MessageBox.Show("Вы ответили правильно", "Верный ответ",

MessageBoxButtons.OK, MessageBoxIcon.Information);

Class1.n += 1;

Class1.mass[8] = 1;

}

else

{

MessageBox.Show("Вы ошиблись", "Неверный ответ", MessageBoxButtons.OK, MessageBoxIcon.Information);

Class1.mass[8] = 0;

}

Vopros10 f = new Vopros10();

this.Hide();

f.Show();

}

private void button2\_Click(object sender, EventArgs e)

{

if (listBox1.SelectedIndex == 0)

{

listBox2.Items.Add(listBox1.Text);

listBox2.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(0);

}

if (listBox1.SelectedIndex == 1)

{

listBox2.Items.Add(listBox1.Text);

listBox2.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(1);

}

if (listBox1.SelectedIndex == 2)

{

listBox2.Items.Add(listBox1.Text);

listBox2.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(2);

}

}

private void button3\_Click(object sender, EventArgs e)

{

if (listBox1.SelectedIndex == 0)

{

listBox3.Items.Add(listBox1.Text);

listBox3.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(0);

}

if (listBox1.SelectedIndex == 1)

{

listBox3.Items.Add(listBox1.Text);

listBox3.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(1);

}

if (listBox1.SelectedIndex == 2)

{

listBox3.Items.Add(listBox1.Text);

listBox3.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(2);

}

}

private void button4\_Click(object sender, EventArgs e)

{

if (listBox1.SelectedIndex == 0)

{

listBox4.Items.Add(listBox1.Text);

listBox4.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(0);

}

if (listBox1.SelectedIndex == 1)

{

listBox4.Items.Add(listBox1.Text);

listBox4.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(1);

}

if (listBox1.SelectedIndex == 2)

{

listBox4.Items.Add(listBox1.Text);

listBox4.Text = Convert.ToString(listBox1.SelectedIndex);

listBox1.Items.RemoveAt(2);

}

}

}

**// Метод для вопросов TrackBar, взят из ЭУ**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros10(trackBar1);

{

Vopros11 f = new Vopros11();

this.Hide();

f.Show();

}

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

label1.Text = String.Format("{0}", trackBar1.Value);

}

}

}

**// Метод для вопросов TrackBar, взят из ЭУ**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros11(trackBar1);

{

Vopros12 f = new Vopros12();

this.Hide();

f.Show();

}

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

label4.Text = String.Format("{0}", trackBar1.Value);

}

}

**// Метод для вопросов ComboBox, взят из ЭУ**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros12(comboBox1, comboBox2, comboBox3);

{

Vopros13 f = new Vopros13();

this.Hide();

f.Show();

}

}

**// Метод для вопросов ComboBox, взят из ЭУ**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros13(comboBox1, comboBox2, comboBox3);

{

Vopros14 f = new Vopros14();

this.Hide();

f.Show();

}

}

**// Метод для вопросов hScrollBar, взят из ЭУ**

private void button1\_Click(object sender, EventArgs e)

{

{

int rez = Class1.Vopros14(hScrollBar1);

{

Vopros15 f = new Vopros15();

this.Hide();

f.Show();

}

}

}

private void hScrollBar1\_Scroll(object sender, ScrollEventArgs e)

{

label2.Text = String.Format("{0}", hScrollBar1.Value);

}

**// Метод для вопросов hScrollBar, взят из ЭУ**

private void button1\_Click(object sender, EventArgs e)

{

{

int rez = Class1.Vopros15(hScrollBar1);

{

Vopros16 f = new Vopros16();

this.Hide();

f.Show();

}

}

}

private void hScrollBar1\_Scroll(object sender, ScrollEventArgs e)

{

label2.Text = String.Format("{0}", hScrollBar1.Value);

}

**// Метод для вопросов CheckedListBox, взят из ЭУ**

private void button1\_Click(object sender, EventArgs e)

{

int rez = Class1.Vopros16(checkedListBox1);

{

Diagramma f = new Diagramma();

this.Hide();

f.Show();

}

}

**// код кнопок для вывода отсортированных результатов теста в datagridview и диаграммы соотношения верных и неверных ответов на второй итоговой форме**

private void button1\_Click(object sender, EventArgs e)

{

chart1.Visible = true;

chart1.Series["Правильно"].Points.AddY(Class1.n);

chart1.Series["Неправильно"].Points.AddY(16 - Class1.n);

}

private void button2\_Click(object sender, EventArgs e)

{

{

Itog f = new Itog();

this.Hide();

f.Show();

}

}

private void button1\_Click(object sender, EventArgs e)

{

label1.Text = "Поздравляем, вы прошли тест. Ваш результат: " + Kursovay\_Rabota.Class1.n.ToString() + " из 16 ";

dataGridView1.Visible = true;

button1.Visible = false;

Kursovay\_Rabota.Class1.VivodDGV(dataGridView1);

}

private void button2\_Click(object sender, EventArgs e)

{

Microsoft.Office.Interop.Excel.Application excelApp = new

Microsoft.Office.Interop.Excel.Application();

excelApp.Visible = true;

var t = Type.Missing;

var Book = excelApp.Workbooks.Add(t);

var Lists = Book.Worksheets;

Worksheet List = Lists.Item[1];

List.Cells[1, 1] = "Результат тестирования";

List.Range["A2", t].Value2 = "Вопрос 1";

List.Range["A3", t].Value2 = "Вопрос 2";

List.Range["A4", t].Value2 = "Вопрос 3";

List.Range["A5", t].Value2 = "Вопрос 4";

List.Range["A6", t].Value2 = "Вопрос 5";

List.Range["A7", t].Value2 = "Вопрос 6";

List.Range["A8", t].Value2 = "Вопрос 7";

List.Range["A9", t].Value2 = "Вопрос 8";

List.Range["A10", t].Value2 = "Вопрос 9";

List.Range["A11", t].Value2 = "Вопрос 10";

List.Range["A12", t].Value2 = "Вопрос 11";

List.Range["A13", t].Value2 = "Вопрос 12";

List.Range["A14", t].Value2 = "Вопрос 13";

List.Range["A15", t].Value2 = "Вопрос 14";

List.Range["A16", t].Value2 = "Вопрос 15";

List.Range["A17", t].Value2 = "Вопрос 16";

List.Range["B1", t].Value2 = "Ответ";

List.Range["B2", t].Value2 = Class1.mass[0];

List.Range["B3", t].Value2 = Class1.mass[1];

List.Range["B4", t].Value2 = Class1.mass[2];

List.Range["B5", t].Value2 = Class1.mass[3];

List.Range["B6", t].Value2 = Class1.mass[4];

List.Range["B7", t].Value2 = Class1.mass[5];

List.Range["B8", t].Value2 = Class1.mass[6];

List.Range["B9", t].Value2 = Class1.mass[7];

List.Range["B10", t].Value2 = Class1.mass[8];

List.Range["B11", t].Value2 = Class1.mass[9];

List.Range["B12", t].Value2 = Class1.mass[10];

List.Range["B13", t].Value2 = Class1.mass[11];

List.Range["B14", t].Value2 = Class1.mass[12];

List.Range["B15", t].Value2 = Class1.mass[13];

List.Range["B16", t].Value2 = Class1.mass[14];

List.Range["B17", t].Value2 = Class1.mass[15];

Microsoft.Office.Interop.Excel.Chart Diagr = excelApp.Charts.Add(t, t, t, t);

Diagr.SetSourceData(List.Range["A2", "B17"],

Microsoft.Office.Interop.Excel.XlRowCol.xlColumns);

Diagr.ChartType = XlChartType.xlColumnClustered;

Diagr.HasLegend = false;

Diagr.HasTitle = true;

Diagr.ChartTitle.Caption = "РЕЗУЛЬТАТЫ ТЕСТА";

Axis Goriz\_Os = Diagr.Axes(Microsoft.Office.Interop.Excel.XlAxisType.xlValue, Microsoft.Office.Interop.Excel.XlAxisGroup.xlPrimary);

Goriz\_Os.HasTitle = true;

Goriz\_Os.AxisTitle.Text = "ОТВЕТЫ";

Axis Vertic\_Os = Diagr.Axes(Microsoft.Office.Interop.Excel.XlAxisType.xlValue, Microsoft.Office.Interop.Excel.XlAxisGroup.xlPrimary);

Vertic\_Os.HasTitle = true;

Vertic\_Os.AxisTitle.Text = "ВОПРОСЫ";

excelApp.ActiveChart.Export(@"C:\Users\evgen\excelfile\Excel.jpg", t, t);

}

private void button3\_Click(object sender, EventArgs e)

{

Class1.Zap\_excel(arr2,15,2);

}

**// Реализация программного кода, которая вызывает появление окна OpenFileDialog с последующим открытием созданного файла.**

private void выводФайлаToolStripMenuItem\_Click(object sender, EventArgs e)

{

OpenFileDialog otkr = new OpenFileDialog();

otkr.DefaultExt = "\*.xlsm;\*.xlsm";

otkr.Filter = "Microsoft Excel (\*xlsm\*) |\*.xls\*";

otkr.Title = "Выберите документ Excel";

if (otkr.ShowDialog() != DialogResult.OK)

{

MessageBox.Show("Вы не выбрали файл", "Открыть", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

System.Diagnostics.Process.Start(otkr.FileName);

}

private void button4\_Click(object sender, EventArgs e)

{

Class1.Bin(dataGridView2);

}

}

}