Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«ЮЖНО-УРАЛЬСКЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(национальный исследовательский университет)

Высшая школа электроники и компьютерных наук

Кафедра «Защиты информации»

Дисциплина

# «Основы управления информационной безопасностью»

ОТЧЕТ

По результатам разработки цифрового продукта (сервиса) для управления осведомленностью (культурой) сотрудников организации в области информационной безопасности.

СОСТАВИЛ

Студент группы КЭ-407

Бесчастнов С.А.

(роспись, фамилия, инициалы)

"\_\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

ПРОВЕРИЛ

Астахова Л.В.

(роспись, фамилия, инициалы)

"\_\_\_\_" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Челябинск

2021 г.

Оглавление

[«Основы управления информационной безопасностью» 1](#_Toc65154592)

[Результаты работы 4](#_Toc65154593)

[Характеристика приложения 4](#_Toc65154594)

[Процесс работы 4](#_Toc65154595)

[Весь предоставляемый функционал 4](#_Toc65154596)

[Положение в системе повышения осведомлённости в организации 8](#_Toc65154597)

[Содержание курсов и их потребитель 8](#_Toc65154598)

[1.КОНФИДЕНЦИАЛЬНОСТЬ 9](#_Toc65154599)

[1.1. Понятие 9](#_Toc65154600)

[1.2 Виды конфиденциальной информации 9](#_Toc65154601)

[1.3 Сущность угрозы 9](#_Toc65154602)

[1.4 Методы защиты 10](#_Toc65154603)

[1.5 Вывод 10](#_Toc65154604)

[2. ЦЕЛОСТНОСТЬ 11](#_Toc65154605)

[2.1. Понятие 11](#_Toc65154606)

[2.2 Сущность угрозы 11](#_Toc65154607)

[2.4 Выводы 11](#_Toc65154608)

[3. ДОСТУПНОСТЬ 12](#_Toc65154609)

[3.1 Понятие 12](#_Toc65154610)

[3.2 Сущность угрозы 12](#_Toc65154611)

[3.3 Методы защиты 12](#_Toc65154612)

[3.4 Вывод 12](#_Toc65154613)

[4 Тесты 13](#_Toc65154614)

[4.1 Вопрос 1 13](#_Toc65154615)

[4.2 Вопрос 2 13](#_Toc65154616)

[4.3 Вопрос 3 13](#_Toc65154617)

[4.4 Вопрос 4 14](#_Toc65154618)

[4.5 Вопрос 5 14](#_Toc65154619)

[4.6 Вопрос 6 14](#_Toc65154620)

[4.7 Вопрос 7 15](#_Toc65154621)

[4.8 Вопрос 8 15](#_Toc65154622)

[4.9 Вопрос 9 15](#_Toc65154623)

[4.10 Вопрос 10 16](#_Toc65154624)

[4.11 Вопрос 11 16](#_Toc65154625)

[4.12 Вопрос 12 16](#_Toc65154626)

[Приложение 1 17](#_Toc65154627)

[Приложение 2 18](#_Toc65154628)

[Приложение 3 19](#_Toc65154629)

[Приложение 4 26](#_Toc65154630)

# **Результаты работы**

В современных условиях требуется постоянно повышать осведомлённости сотрудников всех уровней, в связи с развитием угроз информационной безопасности, с простейшего бытового уровня до минимальных понятий в том числе об целях информационной безопасности.

Назначение программы в системе ИБ организации: повышение осведомлённости сотрудника в сфере информационной безопасности.

# **Характеристика приложения**

Программа не требовательна к системе и может запускаться на любых ЭВМ под управлением ОС Windows и Linux. Программа написана на языке С# .NET Framework 4.8.

Минимальных требований нет в следствии платформы и языка приложения.

Полный код программы (см. Приложение 4).

# **Процесс работы**

Процесс работы с программой выглядит следующим образом:

Сотрудник с некоторой периодичностью проходит курсы и тесты предоставленные в программе тем самым обновляет свои знания в сфере информационной безопасности и так же может оценить уровень своих знаний

# **Весь предоставляемый функционал**

Отображение всех загруженных курсов и оценок за все тесты и максимальный рейтинг по курсу

Доведение тестовых документов

Показ изображений

Проведение тестовых заданий их оценка и отображение оценки

Невозможность сброса результатов теста

По кнопки «Save» cохранение результатов прохождения тестов c полной записью времени и даты сохранения + название каждого курса с общим рейтингом и каждого теста по отдельности. К названию файла автоматически дописывается время и дата сохранения.

При введении в текстовое поле можно сбросить результаты тестирования если ввести команду «reset» или «reset all» для сброса результатов за текущей курса или всех курсов.

Кнопки < и > позволяют пролистывать курсы.

Кнопка ✓ Запускает проверку всех тестов в текущем курсе и блокирует их изменение в пределах одной сессии.

Кнопка 🖌 включает и отключает возможность изменения масштаба элементов с помощью колёсика мыши.

При зажатии ctrl+ колёсико мыши можно изменять размеры текста в текстовых полях.

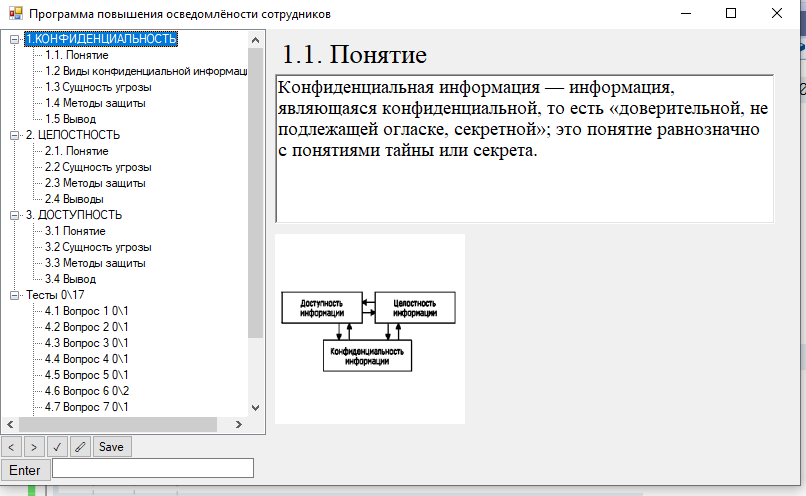


Рисунок 1. Общий вид программы, вывод текста, изображений.

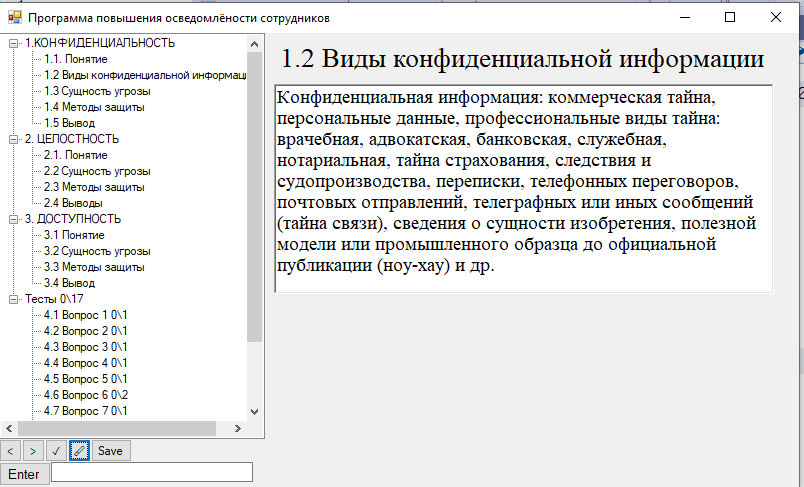


Рисунок 2. Вывод текста.

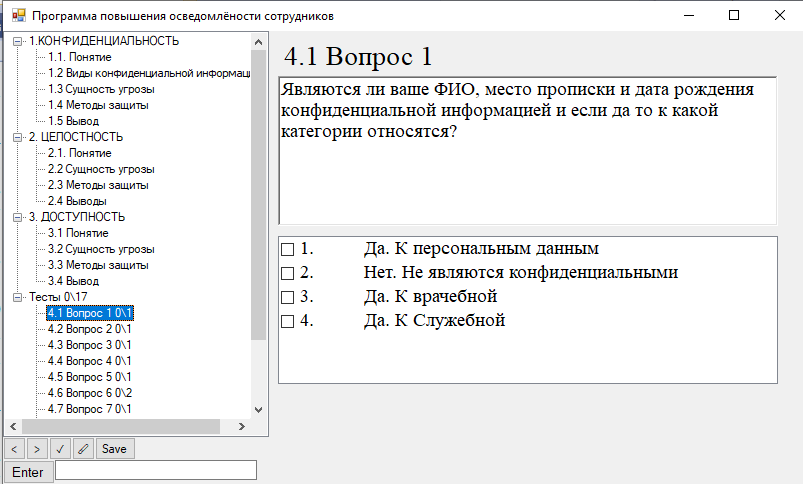


Рисунок 3. Вывод тестовых заданий.

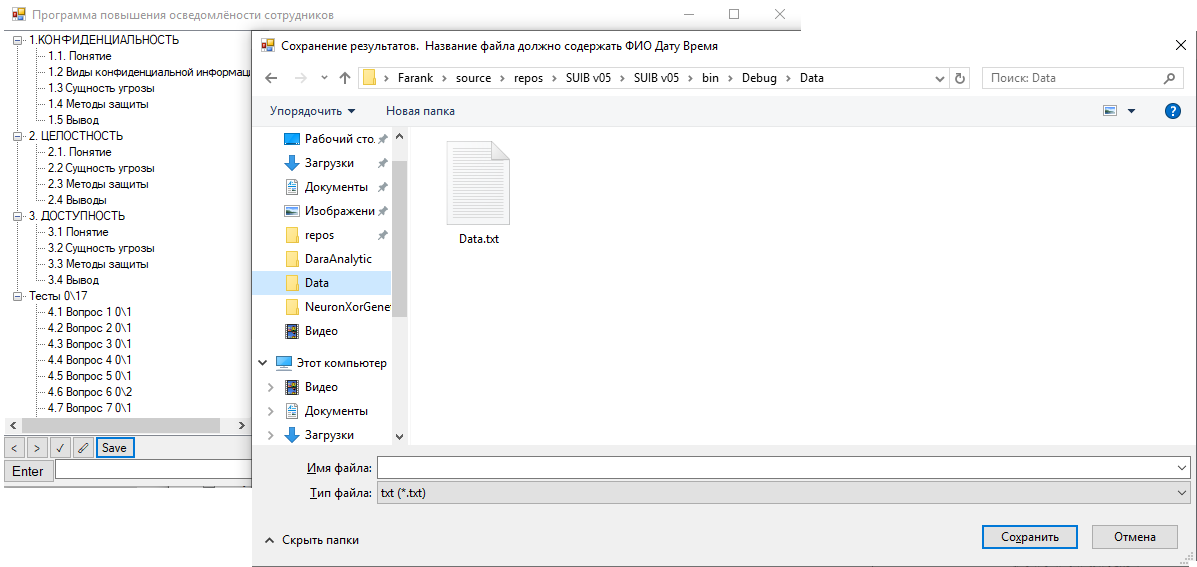


Рисунок 4. Сохранение результатов.

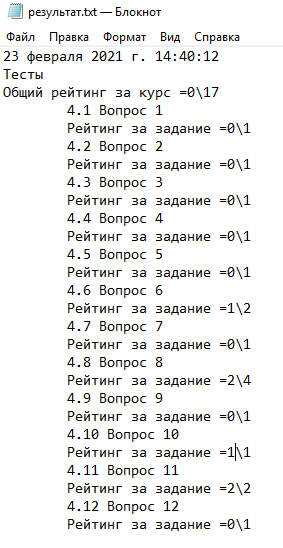


Рисунок 5. Сохранённые результаты.

# **Положение в системе повышения осведомлённости в организации**

При добавлении нового курса или прихода нового сотрудника или введение решения в уже существующую систему порядок функционирования, следующий:

1. Курсы, составленные специалистами ИБ в соответствии с текущей ситуацией в организации размещаются в корне решения с названием «Data.txt» и советующим форматированием (см. Приложение 1).
2. Решение может располагаться на сетевом диске тем самым убирать необходимость загрузки курсов и потом сбора результатов теста с каждого ЭВМ индивидуально.
3. Сотрудник по приказу от руководителя проходит курсы повышая свою осведомленность и решает тесты результат которых сохраняет в файл со своим ФИО по нажатию кнопки «Save» или делает снимок экрана программы. После этого результаты прохождения курсов могут как автоматически собираться для чего предусмотрен предельно упрощенный синтаксис сохраняемых результатов (см. Приложение 2) так и просто просмотрены непосредственным руководителем тем самым давая представление о осведомленности сотрудников в сфере информационной безопасности или какие сотрудники не проявляют должного усилия в изучение материала и поддержания режима информационной безопасности.

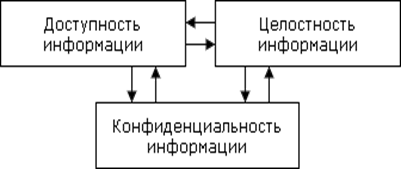
# **Содержание курсов и их потребитель**

Потребитель

1. В составе идущих вместе с программой курсами сотрудники организации не имеющие отношения к службе безопасности и минимально с осведомлённые в сфере ИБ. Содержание курсов в требуемом формате (см. Приложение 3).

Состав приложенных курсов посвящён цели доведения до сотрудника целей информационной безопасности:

1. Конфиденциальность
2. Целостность
3. Доступность



# **1.КОНФИДЕНЦИАЛЬНОСТЬ**

## **1.1. Понятие**

Конфиденциальная информация — информация, являющаяся конфиденциальной, то есть «доверительной, не подлежащей огласке, секретной»; это понятие равнозначно с понятиями тайны или секрета.

## **1.2 Виды конфиденциальной информации**

Конфиденциальная информация: коммерческая тайна, персональные данные, профессиональные виды тайна: врачебная, адвокатская, банковская, служебная, нотариальная, тайна страхования, следствия и судопроизводства, переписки, телефонных переговоров, почтовых отправлений, телеграфных или иных сообщений (тайна связи), сведения о сущности изобретения, полезной модели или промышленного образца до официальной публикации (ноу-хау) и др.

## **1.3 Сущность угрозы**

Угроза нарушения конфиденциальности заключается в том, что информация становится известной тому, кто не располагает полномочиями доступа к ней. Она имеет место, когда получен доступ к некоторой информации ограниченного доступа, хранящейся в вычислительной системе или передаваемой от одной системы к другой. В связи с угрозой нарушения конфиденциальности, используется термин «утечка». Подобные угрозы могут возникать вследствие «человеческого фактора» (например, случайное делегировании тому или иному пользователю привилегий другого пользователя), сбоев работе программных и аппаратных средств.

## **1.4 Методы защиты**

Организационная и просветительная работа с персоналом для донесения важности неразглашения информации персоналом

Технические и программные решения, направленные на доступ к информации только допущенных лиц

## **1.5 Вывод**

Цель обеспечения конфиденциальности доступность информации только авторизованным пользователям, процессам и устройствам.

# **2. ЦЕЛОСТНОСТЬ**

## **2.1. Понятие**

Целостность информации — термин в информатике (криптографии, теории телекоммуникаций, теории информационной безопасности), означающий, что данные не были изменены при выполнении какой-либо операции над ними, будь то передача, хранение или отображение.

## **2.2 Сущность угрозы**

Угрозы целостности (неправомерное изменение данных). Угрозы нарушения целостности — это угрозы, связанные с вероятностью модификации той или иной информации, хранящейся в информационной системе. Нарушение целостности может быть вызвано различными факторами — от умышленных действий персонала до выхода из строя оборудования.

**2.3 Методы защиты**

Обеспечение отказоустойчивости (резервирование, дублирование, зеркалирование оборудования и данных)

Обеспечение безопасного восстановления (резервное копирование и электронное архивирование информации).

## **2.4 Выводы**

 Цель обеспечения целостности – это отсутствие неправомочных искажений, добавлений или уничтожения информации.

# **3. ДОСТУПНОСТЬ**

## **3.1 Понятие**

Доступность информации — состояние информации (ресурсов автоматизированной информационной системы), при котором субъекты, имеющие права доступа, могут реализовывать их беспрепятственно. К правам доступа относятся: право на чтение, изменение, хранение, копирование, уничтожение информации, а также права на изменение, использование, уничтожение ресурсов.

## **3.2 Сущность угрозы**

Угрозы доступности (осуществление действий, делающих невозможным или затрудняющих доступ к ресурсам информационной системы). Нарушение доступности представляет собой создание таких условий, при которых доступ к услуге или информации будет либо заблокирован, либо возможен за время, которое не обеспечит выполнение тех или иных бизнес-целей.

## **3.3 Методы защиты**

Технические, программные и организационные меры, направленные на своевременный доступ всех необходимых лиц и программ к информации

## **3.4 Вывод**

Цель обеспечения доступности — это обеспечение своевременного и надежного доступа к информации и информационным сервисам.

# **4 Тесты**

## **4.1 Вопрос 1**

Вопрос: Являются ли ваше ФИО, место прописки и дата рождения конфиденциальной информацией и если да то к какой категории относятся?

Варианты ответа:

1. Да. К персональным данным
2. Нет. Не являются конфиденциальными
3. Да. К врачебной
4. Да. К Служебной

Верный ответ: 1

## **4.2 Вопрос 2**

Вопрос: Условия входа в организацию через турникеты являются прежде всего решением какой из целей информационной безопасности?

Варианты ответа:

1. Целостность
2. Доступность
3. Конфиденциальность

Верный ответ: 3

## **4.3 Вопрос 3**

Вопрос: Может ли сотрудник повлиять на целостность информации в организации ?

Варианты ответа:

1. Да. Выполняя все надлежащие работы и должностные инструкции, а так же инструкции обращения с оборудованием
2. Нет. Это не его работа
3. Да. Если он сотрудник службы безопасности

Верный ответ: 1

## **4.4 Вопрос 4**

Вопрос: Какие из перечисленных прав являются правами доступа?

Варианты ответа:

1. Право на чтение, изменение
2. Хранение, копирование, уничтожение информации
3. На изменение, хранение
4. Вариант 1 и 2
5. Вариант 2 и 3
6. Вариант 1 и 3
7. Варианты 1,2,3
8. Ни одно

Верный ответ: 7

## **4.5 Вопрос 5**

Вопрос: Какой термин связан с угрозами конфиденциальности?

Варианты ответа:

1. Утечка
2. Кража
3. Потеря

Верный ответ: 1. Утечка

## **4.6 Вопрос 6**

Вопрос: В следствии чего могут возникать угрозы конфиденциальности?

Варианты ответа:

1. Покупка в онлайн магазине
2. Неправильный ввод логина и пароля
3. Нарушение целостности
4. Человеческий фактор
5. Сбои в работе средств

Верный ответ: 4 и 5

## **4.7 Вопрос 7**

Вопрос: В чем состоит цель конфиденциальности?

Варианты ответа:

1. Сохранение информации
2. Обработка информации
3. Допуск только допущенных лиц к информации

Верный ответ: 3

## **4.8 Вопрос 8**

Вопрос: Что из нижеперечисленного относится к угрозам целостности?

Варианты ответа:

1. Модификация
2. Уничтожение
3. Изменение
4. Добавление

Верный ответ: 1,2,3,4

## **4.9 Вопрос 9**

Вопрос: Необходимость предотвращения изменения какой-либо информации – это?

Варианты ответа:

1. Конфиденциальность
2. Целостность
3. Доступность
4. Достоверность

Верный ответ: 2

## **4.10 Вопрос 10**

Вопрос: Какие из угроз относятся к угрозам доступности?

Варианты ответа:

1. Выход из строя дисков с информацией
2. Неправомерный доступ
3. Модификация информации

Верный ответ: 1

## **4.11 Вопрос 11**

Вопрос: Что из перечисленного не относится к числу основных аспектов информационной безопасности?

Варианты ответа:

1. Доступность
2. Целостность
3. Защита от копирования
4. Конфиденциальность
5. Подотчетность

Верный ответ: 3,5

## **4.12 Вопрос 12**

Вопрос: Основная угроза информационной безопасности?

Варианты ответа:

1. Шпионские программы
2. Сетевые черви
3. Троянские программы
4. Человек

Верный ответ: 4

# **Приложение 1**

Образец предоставление курсов для решения в файле «Data.txt»

Курс{

Имя{Имя курса},

Страница{

Имя{Имя страницы },

Текст{Содержание курса},

Картинки{относительный пусть к изображению 1 с именем и расширением, относительный пусть к изображению 2 с именем и расширением },

Ответы{!Верные ответы со знаком восклицательным в начале, не верные без, не верные без }

},

},

# **Приложение 2**

Пример сохранённых результатов тестов в файле «Иванов Иван Иванов 23 фев 2021 14-40-12.txt»

23 февраля 2021 г. 14:40:12

Тесты

Общий рейтинг за курс =0\17

4.1 Вопрос 1

Рейтинг за задание =0\1

4.2 Вопрос 2

Рейтинг за задание =0\1

4.3 Вопрос 3

Рейтинг за задание =0\1

4.4 Вопрос 4

Рейтинг за задание =0\1

4.5 Вопрос 5

Рейтинг за задание =0\1

4.6 Вопрос 6

Рейтинг за задание =0\2

4.7 Вопрос 7

Рейтинг за задание =0\1

4.8 Вопрос 8

Рейтинг за задание =0\4

4.9 Вопрос 9

Рейтинг за задание =0\1

4.10 Вопрос 10

Рейтинг за задание =0\1

4.11 Вопрос 11

Рейтинг за задание =0\2

4.12 Вопрос 12

Рейтинг за задание =0\1

# **Приложение 3**

Содержание файла «Data.txt» с заявленным наполнением и пример синтаксиса

Курс{

Имя{1.КОНФИДЕНЦИАЛЬНОСТЬ},

Страница{

Имя{1.1. Понятие},

Текст{Конфиденциальная информация — информация, являющаяся конфиденциальной, то есть «доверительной, не подлежащей огласке, секретной»; это понятие равнозначно с понятиями тайны или секрета. },

Картинки{Link.bmp},

},

Страница{

Имя{1.2 Виды конфиденциальной информации },

Текст{Конфиденциальная информация: коммерческая тайна, персональные данные, профессиональные виды тайна: врачебная, адвокатская, банковская, служебная, нотариальная, тайна страхования, следствия и судопроизводства, переписки, телефонных переговоров, почтовых отправлений, телеграфных или иных сообщений (тайна связи), сведения о сущности изобретения, полезной модели или промышленного образца до официальной публикации (ноу-хау) и др.}

},

Страница{

Имя{1.3 Сущность угрозы},

Текст{Угроза нарушения конфиденциальности заключается в том, что информация становится известной тому, кто не располагает полномочиями доступа к ней. Она имеет место, когда получен доступ к некоторой информации ограниченного доступа, хранящейся в вычислительной системе или передаваемой от одной системы к другой. В связи с угрозой нарушения конфиденциальности, используется термин «утечка». Подобные угрозы могут возникать вследствие «человеческого фактора» (например, случайное делегировании тому или иному пользователю привилегий другого пользователя), сбоев работе программных и аппаратных средств. }

},

Страница{

Имя{1.4 Методы защиты},

Текст{Организационная и просветительная работа с персоналом для донесения важности неразглашения информации персоналом

Технические и программные решения, направленные на доступ к информации только допущенных лиц

}

},

Страница{

Имя{1.5 Вывод},

Текст{Цель обеспечения конфиденциальности доступность информации только авторизованным пользователям, процессам и устройствам.}

}

},

Курс{

Имя{2. ЦЕЛОСТНОСТЬ},

Страница{

Имя{2.1. Понятие},

Текст{Целостность информации — термин в информатике (криптографии, теории телекоммуникаций, теории информационной безопасности), означающий, что данные не были изменены при выполнении какой-либо операции над ними, будь то передача, хранение или отображение.},

Картинки{Link.bmp},

},

Страница{

Имя{2.2 Сущность угрозы},

Текст{Угрозы целостности (неправомерное изменение данных). Угрозы нарушения целостности — это угрозы, связанные с вероятностью модификации той или иной информации, хранящейся в информационной системе. Нарушение целостности может быть вызвано различными факторами — от умышленных действий персонала до выхода из строя оборудования.}

},

Страница{

Имя{2.3 Методы защиты},

Текст{Обеспечение отказоустойчивости (резервирование, дублирование, зеркалирование оборудования и данных)

Обеспечение безопасного восстановления (резервное копирование и электронное архивирование информации).

}

},

Страница{

Имя{2.4 Выводы},

Текст{Цель обеспечения целостности – это отсутствие неправомочных искажений, добавлений или уничтожения информации. }

}

},

Курс{

Имя{3. ДОСТУПНОСТЬ},

Страница{

Имя{3.1 Понятие},

Текст{Доступность информации — состояние информации (ресурсов автоматизированной информационной системы), при котором субъекты, имеющие права доступа, могут реализовывать их беспрепятственно. К правам доступа относятся: право на чтение, изменение, хранение, копирование, уничтожение информации, а также права на изменение, использование, уничтожение ресурсов.},

Картинки{Link.bmp},

},

Страница{

Имя{3.2 Сущность угрозы},

Текст{Угрозы доступности (осуществление действий, делающих невозможным или затрудняющих доступ к ресурсам информационной системы). Нарушение доступности представляет собой создание таких условий, при которых доступ к услуге или информации будет либо заблокирован, либо возможен за время, которое не обеспечит выполнение тех или иных бизнес-целей.}

},

Страница{

Имя{3.3 Методы защиты},

Текст{Технические, программные и организационные меры, направленные на своевременный доступ всех необходимых лиц и программ к информации}

},

Страница{

Имя{3.4 Вывод},

Текст{Цель обеспечения доступности — это обеспечение своевременного и надежного доступа к информации и информационным сервисам.}

},

}

Курс{

Имя{Тесты},

Страница{

Имя{4.1 Вопрос 1},

Текст{Являются ли ваше ФИО, место прописки и дата рождения конфиденциальной информацией и если да то к какой категории относятся?},

Ответы{!1. Да. К персональным данным,2. Нет. Не являются конфиденциальными,3. Да. К врачебной,4. Да. К Служебной}

},

Страница{

Имя{4.2 Вопрос 2},

Текст{Условия входа в организацию через турникеты являются прежде всего решением какой из целей информационной безопасности?},

Ответы{1. Целостность,2. Доступность,!3. Конфиденциальность}

},

Страница{

Имя{4.3 Вопрос 3},

Текст{Может ли сотрудник повлиять на целостность информации в организации ?},

Ответы{1. Да. Выполняя все надлежащие работы и должностные инструкции, а так же инструкции обращения с оборудованием,2. Нет. Это не его работа,!3. Да. Если он сотрудник службы безопасности}

},

Страница{

Имя{4.4 Вопрос 4},

Текст{Какие из перечисленных прав являются правами доступа?},

Ответы{1. Право на чтение, изменение,2. Хранение, копирование, уничтожение информации,3. На изменение, хранение,4. Вариант 1 и 2,5. Вариант 2 и 3,6. Вариант 1 и 3,!7. Варианты 1,2,3,8. Ни одно}

},

Страница{

Имя{4.5 Вопрос 5},

Текст{Какой термин связан с угрозами конфиденциальности?},

Ответы{!1. Утечка,2. Кража,3. Потеря}

},

Страница{

Имя{4.6 Вопрос 6},

Текст{В следствии чего могут возникать угрозы конфиденциальности?},

Ответы{1. Покупка в онлайн магазине,2. Неправильный ввод логина и пароля,3. Нарушение целостности,!4. Человеческий фактор,!5. Сбои в работе средств}

},

Страница{

Имя{4.7 Вопрос 7},

Текст{В чем состоит цель конфиденциальности?},

Ответы{1. Сохранение информации,2. Обработка информации,!3. Допуск только допущенных лиц к информации}

},

Страница{

Имя{4.8 Вопрос 8},

Текст{Что из нижеперечисленного относится к угрозам целостности?},

Ответы{!1. Модификация ,!2. Уничтожение,!3. Изменение ,!4. Добавление}

},

Страница{

Имя{4.9 Вопрос 9},

Текст{Необходимость предотвращения изменения какой-либо информации – это?},

Ответы{1. Конфиденциальность ,!2. Целостность ,3. Доступность ,4. Достоверность}

},

Страница{

Имя{4.10 Вопрос 10},

Текст{Какие из угроз относятся к угрозам доступности?},

Ответы{!1. Выход из строя дисков с информацией,2. Неправомерный доступ,3. Модификация информации}

},

Страница{

Имя{4.11 Вопрос 11},

Текст{Что из перечисленного не относится к числу основных аспектов информационной безопасности?},

Ответы{1. Доступность,2. Целостность,!3. Защита от копирования,4. Конфиденциальность,!5. Подотчетность}

},

Страница{

Имя{4.12 Вопрос 12},

Текст{Основная угроза информационной безопасности?},

Ответы{1. Шпионские программы,2. Сетевые черви,3. Троянские программы,!4. Человек}

}

},

# **Приложение 4**

Полный текст программы

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace SUIB\_v05

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

Form1.Designer.cs

namespace SUIB\_v05

{

partial class Form1

{

private System.ComponentModel.IContainer components = null;

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

private void InitializeComponent()

{

this.splitContainer1 = new System.Windows.Forms.SplitContainer();

this.button5 = new System.Windows.Forms.Button();

this.textBox1 = new System.Windows.Forms.TextBox();

this.button4 = new System.Windows.Forms.Button();

this.button3 = new System.Windows.Forms.Button();

this.button2 = new System.Windows.Forms.Button();

this.button1 = new System.Windows.Forms.Button();

this.treeView1 = new System.Windows.Forms.TreeView();

this.panel1 = new System.Windows.Forms.Panel();

this.button6 = new System.Windows.Forms.Button();

((System.ComponentModel.ISupportInitialize)(this.splitContainer1)).BeginInit();

this.splitContainer1.Panel1.SuspendLayout();

this.splitContainer1.Panel2.SuspendLayout();

this.splitContainer1.SuspendLayout();

this.SuspendLayout();

//

// splitContainer1

//

this.splitContainer1.Dock = System.Windows.Forms.DockStyle.Fill;

this.splitContainer1.Location = new System.Drawing.Point(0, 0);

this.splitContainer1.Name = "splitContainer1";

//

// splitContainer1.Panel1

//

this.splitContainer1.Panel1.Controls.Add(this.button6);

this.splitContainer1.Panel1.Controls.Add(this.button5);

this.splitContainer1.Panel1.Controls.Add(this.textBox1);

this.splitContainer1.Panel1.Controls.Add(this.button4);

this.splitContainer1.Panel1.Controls.Add(this.button3);

this.splitContainer1.Panel1.Controls.Add(this.button2);

this.splitContainer1.Panel1.Controls.Add(this.button1);

this.splitContainer1.Panel1.Controls.Add(this.treeView1);

//

// splitContainer1.Panel2

//

this.splitContainer1.Panel2.Controls.Add(this.panel1);

this.splitContainer1.Size = new System.Drawing.Size(800, 456);

this.splitContainer1.SplitterDistance = 266;

this.splitContainer1.TabIndex = 0;

//

// button5

//

this.button5.Location = new System.Drawing.Point(87, 406);

this.button5.Name = "button5";

this.button5.Size = new System.Drawing.Size(23, 23);

this.button5.TabIndex = 6;

this.button5.Text = "🖌";

this.button5.UseVisualStyleBackColor = true;

this.button5.Click += new System.EventHandler(this.Button5\_Click);

//

// textBox1

//

this.textBox1.Location = new System.Drawing.Point(61, 427);

this.textBox1.Name = "textBox1";

this.textBox1.Size = new System.Drawing.Size(202, 20);

this.textBox1.TabIndex = 5;

//

// button4

//

this.button4.Font = new System.Drawing.Font("Microsoft Sans Serif", 9.75F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(204)));

this.button4.Location = new System.Drawing.Point(0, 427);

this.button4.Name = "button4";

this.button4.Size = new System.Drawing.Size(52, 24);

this.button4.TabIndex = 4;

this.button4.Text = "Enter";

this.button4.UseVisualStyleBackColor = true;

this.button4.Click += new System.EventHandler(this.Button4\_Click);

//

// button3

//

this.button3.Location = new System.Drawing.Point(58, 406);

this.button3.Name = "button3";

this.button3.Size = new System.Drawing.Size(23, 23);

this.button3.TabIndex = 3;

this.button3.Text = "✓";

this.button3.UseVisualStyleBackColor = true;

this.button3.Click += new System.EventHandler(this.Button3\_Click);

//

// button2

//

this.button2.Location = new System.Drawing.Point(29, 406);

this.button2.Name = "button2";

this.button2.Size = new System.Drawing.Size(23, 23);

this.button2.TabIndex = 2;

this.button2.Text = ">";

this.button2.UseVisualStyleBackColor = true;

this.button2.Click += new System.EventHandler(this.Button2\_Click);

//

// button1

//

this.button1.Location = new System.Drawing.Point(0, 406);

this.button1.Name = "button1";

this.button1.Size = new System.Drawing.Size(23, 23);

this.button1.TabIndex = 1;

this.button1.Text = "<";

this.button1.UseVisualStyleBackColor = true;

this.button1.Click += new System.EventHandler(this.Button1\_Click);

//

// treeView1

//

this.treeView1.Anchor = ((System.Windows.Forms.AnchorStyles)((((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Bottom)

| System.Windows.Forms.AnchorStyles.Left)

| System.Windows.Forms.AnchorStyles.Right)));

this.treeView1.Location = new System.Drawing.Point(0, 0);

this.treeView1.Name = "treeView1";

this.treeView1.Size = new System.Drawing.Size(266, 406);

this.treeView1.TabIndex = 0;

this.treeView1.AfterSelect += new System.Windows.Forms.TreeViewEventHandler(this.TreeView1\_AfterSelect);

//

// panel1

//

this.panel1.AutoScroll = true;

this.panel1.Dock = System.Windows.Forms.DockStyle.Fill;

this.panel1.Location = new System.Drawing.Point(0, 0);

this.panel1.Name = "panel1";

this.panel1.Size = new System.Drawing.Size(530, 456);

this.panel1.TabIndex = 0;

//

// button6

//

this.button6.Location = new System.Drawing.Point(127, 406);

this.button6.Name = "button6";

this.button6.Size = new System.Drawing.Size(41, 23);

this.button6.TabIndex = 7;

this.button6.Text = "Save";

this.button6.UseVisualStyleBackColor = true;

this.button6.Click += new System.EventHandler(this.Button6\_Click);

//

// Form1

//

this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.ClientSize = new System.Drawing.Size(800, 456);

this.Controls.Add(this.splitContainer1);

this.Name = "Form1";

this.Text = "Программа повышения осведомлёности сотрудников ";

this.Load += new System.EventHandler(this.Form1\_Load);

this.ResizeEnd += new System.EventHandler(this.Form1\_ResizeEnd);

this.Enter += new System.EventHandler(this.Form1\_Enter);

this.Resize += new System.EventHandler(this.Form1\_Resize);

this.splitContainer1.Panel1.ResumeLayout(false);

this.splitContainer1.Panel1.PerformLayout();

this.splitContainer1.Panel2.ResumeLayout(false);

((System.ComponentModel.ISupportInitialize)(this.splitContainer1)).EndInit();

this.splitContainer1.ResumeLayout(false);

this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.SplitContainer splitContainer1;

private System.Windows.Forms.TreeView treeView1;

private System.Windows.Forms.Panel panel1;

private System.Windows.Forms.TextBox textBox1;

private System.Windows.Forms.Button button4;

private System.Windows.Forms.Button button3;

private System.Windows.Forms.Button button2;

private System.Windows.Forms.Button button1;

private System.Windows.Forms.Button button5;

private System.Windows.Forms.Button button6;

}

}

Form1.cs

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace SUIB\_v05

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

string path = Path.Combine(Environment.CurrentDirectory, @"Data\", "Data.txt");

if (File.Exists(path))

{

// Create a file to write to.

string Text = "";

Text = File.ReadAllText(path, Encoding.GetEncoding(1251));

parse\_all(Text);

}

draw();

this.MouseWheel += new System.Windows.Forms.MouseEventHandler(this.form\_MouseWheel);

}

public struct skobochki

{

public int pos;

public string frag;

}

private List<skobochki> a = new List<skobochki>();

public void parse\_all(string text)

{

skobochki temp0;

temp0.pos = 0;

temp0.frag = "";

for (int i = 0; i < text.Length; i++)

{

if (text[i] == '{')

{

a.Add(temp0);

temp0.pos++;

temp0.frag = "";

i++;

}

if (text[i] == '}')

{

a.Add(temp0);

temp0.pos--;

temp0.frag = "";

i++;

}

temp0.frag += text[i];

}

/\*

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

{

richTextBox1.Text += "---->" + a[i].frag + "<---- " + Convert.ToString(a[i].pos) + "\n";

}\*/

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

{

if (a[i].pos == 0)

{

course temp1 = new course();

c.Add(temp1);

}

if (i > 0)

{

if (a[i - 1].pos == 1)

{

if (a[i].pos == 2)

{

if (a[i - 1].frag.LastIndexOf("Имя") != -1)

{

if (a[i - 1].frag.Substring(a[i - 1].frag.LastIndexOf("Имя")).Equals("Имя"))

{

c.Last().name = a[i].frag;

}

}

}

}

if (a[i - 1].pos == 2)

{

if (a[i].pos == 3)

{

if (a[i - 1].frag.LastIndexOf("Имя") != -1)

{

if (a[i - 1].frag.Substring(a[i - 1].frag.LastIndexOf("Имя")).Equals("Имя"))

{

course.page temp1 = new course.page();

temp1.name = a[i].frag;

c.Last().p.Add(temp1);

}

}

if (a[i - 1].frag.LastIndexOf("Текст") != -1)

{

if (a[i - 1].frag.Substring(a[i - 1].frag.LastIndexOf("Текст")).Equals("Текст"))

{

c.Last().p.Last().text = a[i].frag;

}

}

if (a[i - 1].frag.LastIndexOf("Картинки") != -1)

{

if (a[i - 1].frag.Substring(a[i - 1].frag.LastIndexOf("Картинки")).Equals("Картинки"))

{

c.Last().p.Last().picture = a[i].frag;

}

}

if (a[i - 1].frag.LastIndexOf("Ответы") != -1)

{

if (a[i - 1].frag.Substring(a[i - 1].frag.LastIndexOf("Ответы")).Equals("Ответы"))

{

c.Last().p.Last().answer = a[i].frag;

}

}

}

}

}

}

/\*

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

{

richTextBox1.Text += c[i].name;

for (int j = 0; j < c[i].p.Count; j++)

{

richTextBox1.Text += c[i].p[j].name;

richTextBox1.Text += c[i].p[j].text;

richTextBox1.Text += c[i].p[j].picture;

richTextBox1.Text += c[i].p[j].answer;

}

}

\*/

analyze();

}

private List<course> c = new List<course>();

private void analyze()

{

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

{

treeView1.Nodes.Add(c[i].name);

c[i].analyze(treeView1.Nodes[treeView1.Nodes.Count - 1]);

c[i].add\_rating();

}

}

public int current\_course = 0;

public int image\_diagonal = 200;

public int textboxsize = 150;

public int answerboxsize = 150;

public class course : Form1

{

public string name = "";

public List<page> p = new List<page>();

public TreeNode this\_tree\_node;

public int current\_page = 0;

public int current\_rating = 0;

public int max\_rating = 0;

public void analyze(TreeNode a)

{

this\_tree\_node = a;

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

{

a.Nodes.Add(p[i].name);

p[i].analyze(a.Nodes[a.Nodes.Count - 1]);

}

}

public void add\_rating()

{

int sum = 0;

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

{

sum += p[i].add\_rating();

}

if (sum > 0)

{

max\_rating = sum;

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

}

public void draw(Panel a)

{

p[current\_page].draw(a);

if (max\_rating > 0)

{

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

else

{

this\_tree\_node.Text = name;

}

}

public void draw(Panel a, int d, int b, int c)

{

p[current\_page].draw(a, d, b, c);

if (max\_rating > 0)

{

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

else

{

this\_tree\_node.Text = name;

}

}

public void change\_treeview\_rating(Panel a, int b)

{

if (max\_rating > 0)

{

p[b].change\_treeview\_rating(a);

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

}

public void chech()

{

p[current\_page].chech();

}

public void chech\_all()

{

int sum = 0;

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

{

sum += p[i].chech(); ;

}

if (sum != 0)

{

current\_rating = sum;

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

}

public void Save(ref string stream)

{

if (max\_rating > 0)

{

stream += "\n" + name;

stream += "\n" + "Общий рейтинг за курс =" + current\_rating + "\\" + max\_rating;

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

{

p[i].Save(ref stream);

}

}

}

public class page : course

{

public string name = "";

public string text = "";

public string picture = "";

public List<string> picture\_path = new List<string>();

public string answer = "";

public TreeNode this\_tree\_node;

public Label nameLabel = new Label();

public RichTextBox textBox = new RichTextBox();

public List<PictureBox> pictureBox = new List<PictureBox>();

public CheckedListBox answerCheckedListBox;

public int current\_rating = 0;

public int max\_rating = 0;

public bool cheched = false;

public List<skobochki> trueanswer = new List<skobochki>();

public void analyze(TreeNode a)

{

this\_tree\_node = a;

if (name != "")

{

nameLabel.Text = name;

nameLabel.Location = new Point(5, 10);

nameLabel.Font = new Font("Times new Roman", 20);

nameLabel.Height = 25;

nameLabel.AutoSize = true;

}

if (text != "")

{

textBox.Text = text;

textBox.Height = textboxsize;

textBox.Width = 300;

textBox.Font = new Font("Times new Roman", 14);

textBox.Location = new Point(5, ((name != "") ? (nameLabel.Location.Y + nameLabel.Height) : 0) + 10);

}

if (picture != "")

{

string[] temp0 = picture.Split(',');

int k = 0;

for (int i = 0; i < temp0.Length; i++)

{

picture\_path.Add(Path.Combine(@"Data\" + temp0[i]));

Bitmap temp2 = new Bitmap(picture\_path.Last());

PictureBox temp3 = new PictureBox();

temp3.Height = image\_diagonal - 10;

temp3.Width = image\_diagonal - 10;

temp3.SizeMode = PictureBoxSizeMode.StretchImage;

temp3.Image = temp2;

temp3.Location = new Point(5 + (i % 3) \* image\_diagonal, ((text != "") ? (textBox.Location.Y + textBox.Height) : ((name != "") ? (nameLabel.Location.Y + nameLabel.Height) : 0)) + ((i > 2) ? (pictureBox[i - 3].Location.Y) : (0)) + 10);

pictureBox.Add(temp3);

if (i >= 1)

{

if (k % 4 == 3)

{

k++;

}

}

}

}

if (answer != "")

{

string[] temp0 = answer.Split(',');

answerCheckedListBox = new CheckedListBox();

answerCheckedListBox.Height = answerboxsize;

answerCheckedListBox.Width = 300;

answerCheckedListBox.Font = new Font("Times new Roman", 14);

answerCheckedListBox.Location = new Point(5, ((picture != "") ? (pictureBox.Last().Location.Y + pictureBox.Last().Height) : ((text != "") ? (textBox.Location.Y + textBox.Height) : ((name != "") ? (nameLabel.Location.Y + nameLabel.Height) : 0))) + 10);

for (int i = 0; i < temp0.Length; i++)

{

if (temp0[i].Contains("!") == true)

{

skobochki temp;

temp.pos = 1;

temp.frag = temp0[i].Replace("!", "");

trueanswer.Add(temp);

}

else

{

skobochki temp;

temp.pos = 0;

temp.frag = temp0[i];

trueanswer.Add(temp);

}

answerCheckedListBox.Items.Add(temp0[i].Replace("!", ""));

}

}

//a.Nodes.Add(text);

}

public void draw(Panel place)

{

GC.Collect();

System.Drawing.Point CurrentPoint; CurrentPoint = place.AutoScrollPosition;

place.SuspendLayout();

place.Controls.Clear();

place.Location = new Point(0, 0);

if (name != "")

{

place.Controls.Add(nameLabel);

}

if (text != "")

{

textBox.Width = place.Width - 30;

//textBox.Height = textboxsize;

place.Controls.Add(textBox);

}

if (picture != "")

{

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

{

//pictureBox[i].Width = image\_diagonal;

//pictureBox[i].Height = image\_diagonal;

place.Controls.Add(pictureBox[i]);

}

}

if (answer != "")

{

answerCheckedListBox.Width = place.Width - 30;

//answerCheckedListBox.Height = answerboxsize;

place.Controls.Add(answerCheckedListBox);

}

place.ResumeLayout();

place.AutoScrollPosition = new Point(Math.Abs(CurrentPoint.X), Math.Abs(place.AutoScrollPosition.Y));

if (max\_rating > 0)

{

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

else

{

this\_tree\_node.Text = name;

}

}

public void draw(Panel place, int a, int b, int c)

{

GC.Collect();

System.Drawing.Point CurrentPoint; CurrentPoint = place.AutoScrollPosition;

place.SuspendLayout();

place.Controls.Clear();

place.Location = new Point(0, 0);

if (name != "")

{

place.Controls.Add(nameLabel);

}

if (text != "")

{

textBox.Width = place.Width - 30;

if (textBox.Height + a > 25)

{

textBox.Height += a;

}

textBox.Location = new Point(5, ((name != "") ? (nameLabel.Location.Y + nameLabel.Height) : 0) + 10);

place.Controls.Add(textBox);

}

if (picture != "")

{

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

{

if (pictureBox[i].Height + a > 25)

{

pictureBox[i].Width += b;

pictureBox[i].Height += b;

}

pictureBox[i].Location = new Point(5 + (i % 3) \* image\_diagonal, ((text != "") ? (textBox.Location.Y + textBox.Height) : ((name != "") ? (nameLabel.Location.Y + nameLabel.Height) : 0)) + ((i > 2) ? (pictureBox[i - 3].Location.Y) : (0)) + 10);

place.Controls.Add(pictureBox[i]);

}

}

if (answer != "")

{

/\*if (answerCheckedListBox.Height + a > 50)

{

answerCheckedListBox.ClientSize = new Size(place.Width - 30, answerCheckedListBox.Height + a);

answerCheckedListBox.Size = answerCheckedListBox.ClientSize;

}\*/

//answerCheckedListBox.Width = place.Width - 30;

answerCheckedListBox.ClientSize = new Size(textBox.Size.Width, textBox.Size.Height);

answerCheckedListBox.Location = new Point(5, ((picture != "") ? (pictureBox.Last().Location.Y + pictureBox.Last().Height) : ((text != "") ? (textBox.Location.Y + textBox.Height) : ((name != "") ? (nameLabel.Location.Y + nameLabel.Height) : 0))) + 10);

place.Controls.Add(answerCheckedListBox);

}

place.ResumeLayout();

place.AutoScrollPosition = new Point(Math.Abs(CurrentPoint.X), Math.Abs(place.AutoScrollPosition.Y));

if (max\_rating > 0)

{

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

else

{

this\_tree\_node.Text = name;

}

}

public void change\_treeview\_rating(Panel a)

{

if (max\_rating > 0)

{

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

}

public int add\_rating()

{

int sum = 0;

if (answer != "")

{

string[] temp0 = answer.Split(',');

for (int i = 0; i < temp0.Length; i++)

{

if (temp0[i].Contains("!") == true)

{

sum++;

}

}

}

if (sum != 0)

{

max\_rating = sum;

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

return sum;

}

public int chech()

{

int true\_ans = 0;

int false\_ans = 0;

int res = 0;

if (max\_rating != 0)

{

if (cheched == false)

{

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

{

if (answerCheckedListBox.GetItemCheckState(i) == CheckState.Checked)

{

if (trueanswer[i].pos == 1)

{

true\_ans++;

}

else

{

false\_ans++;

}

}

}

cheched = true;

//DialogResult result = MessageBox.Show(Convert.ToString(true\_ans) + " " + Convert.ToString(false\_ans), "Сочувствие", MessageBoxButtons.OK, MessageBoxIcon.Information, MessageBoxDefaultButton.Button1);

if (true\_ans == max\_rating && false\_ans == 0)

{

current\_rating = max\_rating;

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

if (true\_ans < max\_rating && false\_ans == 0)

{

current\_rating = true\_ans;

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

if (false\_ans > 0)

{

current\_rating = 0;

this\_tree\_node.Text = name + " " + current\_rating + "\\" + max\_rating;

}

}

}

return current\_rating;

}

public void Save(ref string stream)

{

if (max\_rating > 0)

{

stream += "\n\t" + name;

stream += "\n\t" + "Рейтинг за задание =" + current\_rating + "\\" + max\_rating;

}

}

}

}

public void draw()

{

c[current\_course].draw(panel1);

draw\_button();

}

public void drawall()

{

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

{

for (int j = 0; j < c[i].p.Count; j++)

{

c[i].change\_treeview\_rating(panel1, j);

}

}

c[current\_course].draw(panel1);

draw\_button();

}

public void draw\_button()

{

button1.Location = new Point(0, treeView1.Location.Y + treeView1.Height);

button2.Location = new Point(button1.Location.X + button1.Width, treeView1.Location.Y + treeView1.Height);

button3.Location = new Point(button2.Location.X + button2.Width, treeView1.Location.Y + treeView1.Height);

button5.Location = new Point(button3.Location.X + button3.Width, treeView1.Location.Y + treeView1.Height);

button6.Location = new Point(button5.Location.X + button5.Width, treeView1.Location.Y + treeView1.Height);

button4.Location = new Point(0, button1.Location.Y + button1.Height);

textBox1.Location = new Point(button4.Location.X + button4.Width, button1.Location.Y + button1.Height);

}

private void TreeView1\_AfterSelect(object sender, TreeViewEventArgs e)

{

if (treeView1.SelectedNode.Parent != null)

{

//c[treeView1.SelectedNode.Index].draw\_all(panel1, 0);

current\_course = treeView1.SelectedNode.Parent.Index;

c[current\_course].current\_page = treeView1.SelectedNode.Index;

draw();

}

else

{

current\_course = treeView1.SelectedNode.Index;

draw();

}

}

private void Form1\_ResizeEnd(object sender, EventArgs e)

{

draw();

}

private void Form1\_Enter(object sender, EventArgs e)

{

draw();

}

private void Form1\_Resize(object sender, EventArgs e)

{

draw();

}

private void Button1\_Click(object sender, EventArgs e)

{

if (c[current\_course].current\_page > 0)

{

//c[current\_course].chech();

c[current\_course].current\_page--;

}

else

{

if (current\_course > 0)

{

c[current\_course].chech\_all();

current\_course--;

c[current\_course].current\_page = c[current\_course].p.Count - 1;

}

}

draw();

}

private void Button2\_Click(object sender, EventArgs e)

{

if (c[current\_course].current\_page < c[current\_course].p.Count - 1)

{

//c[current\_course].chech();

c[current\_course].current\_page++;

}

else

{

if (current\_course < c.Count - 1)

{

c[current\_course].chech\_all();

//MessageBox.Show("Курс завершен\nРезультаты за курс\n" + Convert.ToString(c[current\_course].current\_rating) + "\\" + Convert.ToString(c[current\_course].max\_rating), "Курс завершен", MessageBoxButtons.OK, MessageBoxIcon.Information, MessageBoxDefaultButton.Button1);

current\_course++;

c[current\_course].current\_page = 0;

}

}

draw();

}

private void Button3\_Click(object sender, EventArgs e)

{

c[current\_course].chech\_all();

}

private void Button4\_Click(object sender, EventArgs e)

{

if (textBox1.Text == "reset" || textBox1.Text == "Reset")

{

c[current\_course].p[c[current\_course].current\_page].cheched = false;

c[current\_course].current\_rating -= c[current\_course].p[c[current\_course].current\_page].current\_rating;

c[current\_course].p[c[current\_course].current\_page].current\_rating = 0;

drawall();

textBox1.Text = "";

}

if (textBox1.Text == "reset all" || textBox1.Text == "Reset all")

{

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

{

for (int j = 0; j < c[i].p.Count; j++)

{

c[i].p[j].cheched = false;

c[i].current\_rating -= c[i].p[j].current\_rating;

c[i].p[j].current\_rating = 0;

}

}

drawall();

textBox1.Text = "";

}

}

private int resize = 0;

private void Button5\_Click(object sender, EventArgs e)

{

if (resize == 0)

{

resize = 1;

}

else

{

resize = 0;

}

}

private void form\_MouseWheel(object sender, System.Windows.Forms.MouseEventArgs e)

{

if (resize == 1)

{

c[current\_course].draw(panel1, e.Delta / 10, e.Delta / 10, e.Delta / 10);

//draw();

}

}

private void Button6\_Click(object sender, EventArgs e)

{

SaveFileDialog saveFileDialog1 = new SaveFileDialog();

saveFileDialog1.Filter = "txt|\*.txt";

saveFileDialog1.Title = "Сохранение результатов. \n Название файла должно содержать ФИО Дату Время";

saveFileDialog1.ShowDialog();

if (saveFileDialog1.FileName != "")

{

DateTime now = DateTime.Now;

string save\_to\_file = now.ToString("F");

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

{

c[i].Save(ref save\_to\_file);

}

File.WriteAllText(saveFileDialog1.FileName, save\_to\_file);

}

}

}

}