Федеральное государственное бюджетное профессиональное образовательное учреждение

Торжокский политехнический колледж

Федерального агентства по государственным резервам

Зимнее задание по МДК.01.01

«Разработка программных модулей»

Выполнил студент группы П-30

Гаврильченко С.А.

Баранов Д.М.

2025

**Раздел 1 – Практическая значимость программы**

Требования к программному модулю:

1. Структура хранения вопросов:

- Необходимо создать структуру данных для хранения вопросов в трех разделах: «знать», «уметь», «владеть».

- Каждый вопрос должен быть уникальным и иметь идентификатор для удобного доступа.

2. Интерфейс пользователя:

- Пользователь должен иметь возможность ввести количество необходимых билетов.

- Предусмотреть возможность выбора файла для сохранения билетов и файла с вопросами.

3. Случайный отбор вопросов:

- Модуль должен реализовать алгоритм для случайного выбора вопросов из каждой категории, гарантируя, что выбранные вопросы не повторяются в пределах одного билета.

- Обеспечить, чтобы каждый билет содержал ровно по одному вопросу из каждого раздела.

4. Проверка наличия достаточного количества вопросов:

- Перед созданием билетов необходимо проверить, достаточно ли вопросов в каждой категории для выполнения задания.

- Если вопросов оказывается недостаточно, система должна информировать пользователя об этом.

5. Добавление новых вопросов:

- В случае нехватки вопросов должна быть предусмотрена возможность ввода дополнительных вопросов вручную.

- Обновление исходного текстового файла с вопросами должно происходить автоматически после ввода новых данных.

6. Сохранение и форматирование билетов:

- Сохранение сгенерированных экзаменационных билетов в текстовом файле должно быть реализовано с учётом формата, понятного пользователю.

- Каждый билет должен быть четко структурирован, чтобы пользователь мог легко его читать.

7. Обработка ошибок:

- Модуль должен обрабатывать возможные ошибки, такие как неверный ввод пользователя, отсутствие доступа к файлам и т.д.

- Предусмотреть возможность повторного запроса ввода в случае ошибок.

8. Кросс-платформенность:

- Модуль должен быть разработан с учётом возможности использования на различных операционных системах.

Проблемы, которые может решить программа:

1. Автоматизация процесса создания экзаменационных билетов:

- Программа значительно упрощает подготовку билетов, избавляя преподавателя от необходимости вручную отбирать вопросы.

2. Сокращение времени на составление билетов:

- Быстрое и эффективное генерирование экзаменационных билетов позволяет преподавателю сэкономить время и сосредоточиться на других аспектах подготовки к экзамену.

3. Предотвращение повторения вопросов:

- Алгоритм обеспечивает отсутствие дублирующихся вопросов в билетах.

4. Разбиение по разделам:

- Возможность выбора вопросов по различным разделам помогает равномерно оценить знания студентов и разнообразит экзаменационный процесс.

5. Генерация случайных вопросов:

- Студенты не могут предугадать, какие вопросы будут в экзаменационных билетах, что увеличивает объективность оценки.

6. Сохранение и управление вопросами:

- Программа дает возможность дополнительно вводить и обновлять вопросы, что позволяет поддерживать актуальность содержания экзамена.

7. Уведомление об недостатке вопросов:

- Программа информирует преподавателя о необходимости добавления новых вопросов.

8. Легкость в экспорте данных:

- Возможность сохранения сгенерированных билетов в текстовый файл позволяет преподавателю легко распечатать.

9. Гибкость в настройках:

- Пользователь может самостоятельно определить количество билетов, что позволяет адаптировать программу под различные требования и условия.

**Раздел 2 – Функциональные требования**

1. Пользователь определяет и вводит желаемое количество экзаменационных билетов.

2. Программа автоматически выбирает случайный вопрос из трех разделов: «знать», «уметь», «владеть» для создания билета.

3. Перед созданием билетов программа проверяет, достаточно ли вопросов в каждом разделе для запрашиваемого количества билетов.

4. Если вопросов недостаточно, программа уведомляет пользователя об этом и указывает какие разделы требуется дополнить.

5. Программа предлагает пользователю ввести дополнительные вопросы для каждого из разделов, где не хватает материала, с последующим обновлением исходного текстового файла.

6. После генерации экзаменационных билетов программа сохраняет их в текстовый файл, имя и местоположение которого указывает пользователь.

7. Программа гарантирует, что в каждом из сгенерированных билетов вопросы из разных разделов не повторяются.

8. Программа отображает сгенерированные билеты непосредственно в консоли для предварительного просмотра пользователем перед сохранением.

**Раздел 3 – Проектирование модульной структуры программы**

В данной программе представлены 4 класса.

Класс Question отвечает за представление вопроса, т.е. представляет единичный вопрос для экзаменационного билета.

|  |
| --- |
| Question |
| public string Text { get; set; }  public string Section { get; set; } |
| - |

Класс Ticket отвечает за представление билета со списком вопросов, т.е. представляет экзаменационный билет, который состоит из набора вопросов.

|  |
| --- |
| Ticket |
| public List<Question> Questions { get; set; } = new List<Question>(); |
| public override string ToString() |

Класс QuestionManager управляет списком всех вопросов, загружает их из файла, добавляет новые вопросы и предоставляет возможность выбрать случайный вопрос из определенного раздела.

|  |
| --- |
| QuestionManager |
| \_filename = filename;  LoadQuestionsFromFile(); |
| public void LoadQuestionsFromFile()  public void AddQuestion(string text, string section)  public Question GetRandomQuestion(string section, List<Question> usedQuestions)  public bool HasEnoughQuestions(int numTickets) |

Класс TicketGenerator отвечает за создание экзаменационных билетов на основе имеющихся вопросов.

|  |
| --- |
| TicketGenerator |
| - |
| **public** List<Ticket> **GenerateTickets**(QuestionManager questionManager, int numTickets) |

**Раздел 4 – Описание алгоритмов программы.**

Метод LoadQuestionsFromFile() отвечает за чтение вопросов из файла и создания объектов Question (вопрос), которые впоследствии добавляет в список Questions (вопросы).

Работает по следующему принципу:

* Читает вопросы из указанного файла.
* Разделяет каждую строку на раздел и текст вопроса, используя символ |.
* Создает объекты Question и добавляет их в список Questions.
* Очищает список Questions перед загрузкой новых вопросов

Метод AddQuestion(string text, string section) отвечает за добавление нового объекта Question (вопрос).

Работает по следующему принципу:

* Создает новый объект Question.
* Добавляет новый вопрос в список Questions.
* Сохраняет вопрос в файл