Программный модуль «Учет успеваемости студентов». Программный модуль предназначен для оперативного учета успеваемости студентов в сессию заместителем директора по СПО и учебной частью. Сведения об успеваемости студентов должны храниться в течение всего срока их обучения и использоваться при составлении справок о прослушанных курсах и приложений к диплому.

**1. Общие сведения**

Полное наименование системы:

Программный модуль "Учет успеваемости студентов".

Условное обозначение системы:

ПМ " Учет успеваемости студентов"

Программное обеспечение, необходимое для функционирования программы:

* Операционная система Windows
* Система управления базами данных PostgreSQL

Языки программирования, на которых написана программа:

* Python
* Фреймворк Tkinter
* Фреймворк Django

**2. Функциональное назначение**

Классы решаемых задач:

Программный модуль "Учет успеваемости студентов" решает следующие классы задач:

1. Учет успеваемости студентов: Модуль позволяет вести записи о результате сдачи экзаменов, зачетов и других учебных мероприятий для каждого студента. В эти записи могут входить оценки, даты сдачи, преподаватели и другие связанные с данным событием данные.
2. Генерация отчетов: Модуль предоставляет функциональность для генерации отчетов о успеваемости студентов, которые могут быть использованы административным персоналом для различных целей, включая подготовку справок о прослушанных курсах и приложений к диплому.

Назначение программы:

Основное назначение программы заключается в обеспечении учета успеваемости студентов и обеспечении доступа к соответствующей информации для административных сотрудников учебного учреждения. Это позволяет эффективно управлять информацией о студентах, их оценках и прогрессе в течение всего срока обучения. Кроме того, программа упрощает процесс подготовки документов, связанных с учебным процессом.

Сведения о функциональных ограничениях на применение:

* Доступ к модулю ограничен административным сотрудникам, ответственным за учет успеваемости студентов, и не предоставляется студентам или другим пользователям.
* Модуль может ограничивать доступ к изменению и удалению данных успеваемости, чтобы избежать нежелательных изменений или утери информации.
* Модуль может иметь ограниченные права доступа для каждого пользователя в зависимости от их роли и полномочий в учебном учреждении.
* Безопасность и конфиденциальность данных студентов должны быть обеспечены согласно законодательству и политикам учебного учреждения.

Эти ограничения помогают защитить конфиденциальность данных и обеспечить правильное функционирование программного модуля.

**3. Описание логической структуры**

Алгоритм программы:

Программный модуль "Учет успеваемости студентов" разделен на следующие ключевые компоненты и алгоритмы:

1. Вход и аутентификация:
   * Пользователи, имеющие доступ к модулю, аутентифицируются с использованием своих учетных данных (логин и пароль) или других методов аутентификации, если они реализованы.
2. Управление данными студентов:
   * Создание, редактирование и удаление записей о студентах, включая их личные данные и идентификационные номера.
   * Ведение информации о курсах, которые студенты должны посещать.
3. Учет успеваемости:
   * Запись оценок, дат сдачи экзаменов и другой информации о успеваемости студентов.
4. Генерация отчетов:
   * Система позволяет административным сотрудникам создавать различные отчеты о успеваемости студентов. Эти отчеты могут включать информацию о прослушанных курсах, оценках, статистике успеваемости и другие данные.

Используемые методы:

* Методы для работы с базой данных, такие как добавление, обновление, удаление и выборка данных.
* Алгоритмы аутентификации и авторизации пользователей.
* Методы для генерации отчетов и экспорта данных в различные форматы, такие как PDF, Excel и другие.
* Методы валидации данных, чтобы гарантировать целостность и правильность введенных данных.

Структура программы с описанием функций составных частей и связи между ними:

1. Интерфейс пользователя:
   * Обеспечивает взаимодействие пользователей с программой. Включает в себя формы для ввода данных, кнопки и другие элементы управления.
2. Модуль управления студентами:
   * Содержит функции для добавления, редактирования и удаления записей о студентах, а также связанных с ними курсах.
3. Модуль учета успеваемости:
   * Содержит функции для ведения записей об успеваемости студентов, включая оценки и даты сдачи экзаменов.
4. Модуль генерации отчетов:
   * Отвечает за создание отчетов на основе данных из базы данных. Может быть интегрирован с системами генерации документов, такими как LaTeX или Microsoft Word.
5. Модуль базы данных:
   * Хранит данные о студентах, их успеваемости и другие связанные данные.

Эти компоненты взаимодействуют между собой, обеспечивая целостность и функциональность программного модуля.

Связи программы с другими программами:

1. Интеграция с системой управления учебным процессом:
   * Модуль может интегрироваться с системой, отслеживающей расписание занятий, аудитории и другие аспекты учебного процесса. Это позволит автоматически связывать информацию о курсах, экзаменах и успеваемости студентов с расписанием.
2. Импорт и экспорт данных:
   * Модуль может поддерживать импорт и экспорт данных в различных форматах, таких как CSV или Excel. Это позволит обмениваться информацией между программой "Учет успеваемости студентов" и другими приложениями или системами в учебном учреждении.
3. Интеграция с системой аутентификации:
   * Модуль может использовать существующую систему аутентификации учебного учреждения, чтобы обеспечить безопасный доступ для пользователей. Это позволит избежать необходимости создания отдельных учетных записей для каждого пользователя.
4. Интеграция с системой электронного документооборота:
   * Модуль может интегрироваться с системой управления документами, что упростит процесс создания и хранения документов, таких как справки о прослушанных курсах и приложения к диплому.
5. Интеграция с системой учета финансов:
   * Если учебное учреждение ведет финансовый учет, модуль может интегрироваться с системой учета финансов для автоматического учета платежей и сбора данных о задолженностях студентов.

**4. Используемые технические средства**

1. Сервер:
   * Физический сервер или виртуальная машина (VM) с достаточными ресурсами, такими как процессоры, оперативная память и хранилище. Например, сервер с 4 ядрами CPU, 16 ГБ оперативной памяти и 100 ГБ SSD-накопителем.
2. Операционная система:
   * Windows Server 2019.
3. Система управления базами данных (СУБД):
   * PostgreSQL 13.0 или выше.
4. Веб-сервер (при использовании веб-интерфейса):
   * Nginx для проксирования запросов к приложению.
5. Язык программирования:
   * Python 3.x.
6. Интегрированная среда разработки (IDE):
   * Visual Studio Code для разработки и отладки кода.
7. Браузеры:
   * Поддерживаемый браузеры для веб-интерфейса Chrome.
8. Сетевое оборудование:
   * Маршрутизатор, коммутатор и средства обеспечения безопасности сети, включая брандмауэр.
9. Средства резервного копирования:
   * Регулярное резервное копирование данных на внешние накопители или удаленные серверы.
10. Средства безопасности:
    * Брандмауэр, система обнаружения вторжений, шифрование данных в покое и в передаче (HTTPS).
11. Средства мониторинга и управления системой:
    * Мониторинг сервера с использованием инструмента Nagios и система управления конфигурацией Ansible.

**5. Вызов и загрузка**

Способ вызова программы с соответствующего носителя данных:

Веб-интерфейс: Пользователи могут вызывать программу, открыв веб-браузер и перейдя по адресу (URL) веб-интерфейса, где они могут вводить данные и выполнять необходимые действия.

Входные точки в программу:

Формы веб-интерфейса: На веб-страницах пользователи могут вводить информацию о студентах, курсах и оценках.

Импорт данных: Программа может поддерживать импорт данных из файлов, таких как CSV, Excel или других форматов.

**6. Входные данные**

Характер, организация и предварительная подготовка входных данных

Входные данные включают в себя информацию о студентах, учебных курсах, оценках, датах сдачи экзаменов и другие связанные данные. Эти данные организованы в следующие категории:

1. Данные студентов:
   * Имя, фамилия, дата рождения, контактные данные студентов.
   * Идентификационные номера студентов, такие как учебные номера.
   * Дополнительные характеристики, такие как факультет, курс, группа и т. д.
2. Данные учебных курсов:
   * Название курса, код курса, описание.
   * Информация о преподавателях и аудиториях, где проводятся занятия.
3. Данные оценок и успеваемости:
   * Оценки студентов, полученные на экзаменах или зачетах.
   * Даты сдачи экзаменов и другие мероприятия.
4. Дополнительные сведения:
   * Другие данные, такие как замечания, справки и документы о студентах.

Входные данные должны быть предварительно подготовлены и структурированы, чтобы их можно было корректно ввести в программу. Это включает в себя проверку данных на наличие ошибок, очистку данных от лишних символов и форматирование данных в соответствии с ожидаемой структурой.

Формат, описание и способ кодирования входных данных:

1. Текстовые данные:
   * Формат: Текстовые данные представлены в виде plain text (обычный текст), CSV (Comma-Separated Values), JSON (JavaScript Object Notation) и XML (eXtensible Markup Language).
   * Кодирование: Данные закодированы в UTF-16 для поддержки разных языков и символов.
2. Документы и изображения:
   * Формат: PDF (Portable Document Format), DOCX (Microsoft Word), XLSX (Microsoft Excel) и изображения в форматах JPEG, PNG
   * Кодирование: Данные хранятся в бинарном виде.
3. Данные из базы данных:
   * Формат: Данные из СУБД извлекаются в структурированном виде с использованием SQL-запросов.
   * Кодирование: Для передачи данных между программами и СУБД используется соответствующий протокол JDBC.
4. Данные через веб-службы (API):
   * Формат: Для передачи данных через веб-службы используется формат JSON.
   * Кодирование: Данные передаются через HTTP с использованием POST.

**7. Выходные данные**

Характер и организация выходных данных

Выходные данные представляют собой информацию, которая формируется на основе входных данных и действий пользователя. Характер и организация выходных данных включают в себя следующие элементы:

1. Отчеты и справки:
   * Программа может генерировать отчеты и справки о прослушанных курсах, оценках, успеваемости студентов и других статистических данных.
2. Экспорт данных:
   * Возможность экспорта данных в различные форматы, такие как PDF, Excel, CSV или другие, чтобы пользователи могли сохранить данные на своих устройствах.
3. Веб-интерфейс:
   * Если программа имеет веб-интерфейс, то выходные данные отображаются в браузере пользователя. Это могут быть динамические страницы с информацией о студентах, курсах и оценках.
4. API-ответы:
   * В случае использования API, выходные данные представляют собой ответы на API-запросы, предоставляя структурированные данные для других приложений.
5. Документы и файлы:
   * Программа может создавать документы, такие как справки о прослушанных курсах, которые могут быть сохранены как файлы в определенном формате.

Формат, описание и способ кодирования выходных данных

1. Текстовые данные:
   * Формат: Данные представлены в формате plain text или в форматах, таких как HTML для веб-страниц.
   * Кодирование: Данные могут быть закодированы в UTF-16.
2. Документы и изображения:
   * Формат: Документы могут быть в форматах, таких как PDF, DOCX, XLSX, и изображения в форматах JPEG, PNG.
   * Кодирование: Данные хранятся в бинарном виде.
3. Данные из базы данных:
   * Формат: в структурированном виде с использованием SQL-запросов.
   * Кодирование: Данные представлены в стандартном формате базы данных.
4. Данные через веб-службы (API):
   * Формат: Для передачи данных через веб-службы используется JSON.
   * Кодирование: Данные передаются через HTTP с использованием POST.

Формат и способ кодирования данных могут быть адаптированы под нужды пользователей и интегрирующих приложений, чтобы обеспечить удобство использования и обмен данных.