
Класс User:

Поля:

login (строка) - логин пользователя

password (строка) - пароль пользователя

access_rights (строка) - права доступа пользователя

Методы:

get_login() - возвращает логин пользователя

set_login(login) - устанавливает логин пользователя

get_password() - возвращает пароль пользователя

set_password(password) - устанавливает пароль пользователя

get_access_rights() - возвращает права доступа пользователя

set_access_rights(access_rights) - устанавливает права доступа пользователя

Класс Teacher:

Наследует от класса **User**

Поля:

courses (список) - список курсов, которые ведет преподаватель

student_groups (список) - список групп студентов, которыми занимается преподаватель

Методы:

get_courses() - возвращает список курсов, которые ведет преподаватель

set_courses(courses) - устанавливает список курсов, которые ведет преподаватель

get_student_groups() - возвращает список групп студентов, которыми занимается преподаватель

set_student_groups(student_groups) - устанавливает список групп студентов, которыми занимается преподаватель

Класс Course:

Поля:

course_id (число) - идентификатор курса
course_name (строка) - название курса
course_description (строка) - описание курса
//start_date (дата) - дата начала курса
//end_date (дата) - дата окончания курса
tests (список) - список тестов курса
students (список) - список студентов, записавшихся на курс

Методы:

add_test():

Этот метод добавляет новый тест в список тестов курса. Метод принимает объект Test и добавляет его в список tests. Метод не возвращает никакого значения.

remove_test():

Этот метод удаляет тест из списка тестов курса. Метод принимает объект Test и удаляет его из списка tests. Метод не возвращает никакого значения.

get_tests():

Этот метод возвращает список тестов курса.

add_student():

Этот метод добавляет студента в список студентов курса. Метод принимает объект Student и добавляет его в список students. Метод не возвращает никакого значения.

remove_student():

Этот метод удаляет студента из списка студентов курса. Метод принимает объект Student и удаляет его из списка students. Метод не возвращает никакого значения.

get_students():

Этот метод возвращает список студентов курса.

get_course_id():

Этот метод возвращает идентификатор курса.

set_course_name():

Этот метод устанавливает название курса. Метод принимает строку `course_name` и устанавливает ее в соответствующее поле объекта. Метод не возвращает никакого значения.

`get_course_name()`:

Этот метод возвращает название курса.

Класс Authentication:

Алгоритм входа в систему:

Проверка, не авторизован ли пользователь уже в системе. Если пользователь уже авторизован, выдать сообщение об этом и прервать процесс входа в систему.

Запросить у пользователя логин и пароль.

Проверить корректность введенных данных, сравнив их с данными из базы данных.

Если данные корректны, создать сессию для пользователя и сохранить ее в базе данных. Если данные некорректны, выдать соответствующее сообщение и прервать процесс входа в систему.

Алгоритм выхода из системы:

Проверить, авторизован ли пользователь в системе. Если пользователь не авторизован, выдать сообщение об этом и прервать процесс выхода из системы.

Удалить сессию пользователя из базы данных.

Выдать сообщение о успешном выходе из системы.

Методы:

`login()`: вход в систему.

`logout()`: выход из системы.

Класс TestAdministrator:

Метод `create_test()`:

Создает новый тест с заданными параметрами (название, дата, время и т.д.)

Добавляет тест в хранилище тестов

Возвращает созданный тест

Метод add_question():

Добавляет новый вопрос в указанный тест

Возвращает результат выполнения метода (успех/ошибка)

Метод remove_question():

Удаляет указанный вопрос из теста

Если вопрос не найден, выводит сообщение об ошибке

Возвращает результат выполнения метода (успех/ошибка)

Метод update_question():

Обновляет информацию о вопросе (например, ответы на вопрос и правильный ответ)

Если вопрос не найден, выводит сообщение об ошибке

Возвращает результат выполнения метода (успех/ошибка)

Метод assign_test_to_group():

Назначает тест указанной группе

Возвращает результат выполнения метода (успех/ошибка)

Метод assign_test_to_student():

Назначает тест указанному студенту

Возвращает результат выполнения метода (успех/ошибка)

Метод change_test_date():

Проверить, что дата и время, указанные для изменения, корректны и не конфликтуют с другими запланированными событиями.

Изменить дату и время теста в базе данных или другом источнике хранения информации.

Метод delete_test():

Проверить, что тест был назначен и еще не прошел.

Удалить информацию о тесте из базы данных

Класс Test:

Метод add_question():

Выдает приглашение для ввода вопроса и вариантов ответа

Выполняет ввод вопроса и вариантов ответа

Выполняет контроль корректности введенных данных

Добавляет вопрос и ответы в список вопросов и ответов

Если вопрос уже существует, то метод обновляет ответы на новые

Возвращает результат выполнения метода (успех/ошибка)

Метод remove_question():

Выдает приглашение для ввода вопроса, который нужно удалить

Выполняет ввод вопроса

Ищет вопрос в списке вопросов и ответов

Если вопрос найден, удаляет его из списка вопросов и ответов

Если вопрос не найден, выводит сообщение об ошибке

Возвращает результат выполнения метода (успех/ошибка)

Метод update_question():

Выдает приглашение для ввода вопроса, который нужно обновить

Выполняет ввод вопроса

Ищет вопрос в списке вопросов и ответов

Если вопрос найден, обновляет ответы и правильный ответ на новые

Если вопрос не найден, выводит сообщение об ошибке

Возвращает результат выполнения метода (успех/ошибка)

Класс Student:

Метод add_student():

Создает новый объект Student с именем и статистикой по тестам

Добавляет нового студента в список студентов

Возвращает результат выполнения метода (успех/ошибка)

Метод remove_student():

Выдает приглашение для ввода имени студента, которого нужно удалить

Выполняет ввод имени

Ищет студента с таким именем в списке студентов

Если студент найден, удаляет его из списка студентов

Если студент не найден, выводит сообщение об ошибке

Возвращает результат выполнения метода (успех/ошибка)

Метод update_student():

Выдает приглашение для ввода имени студента, которого нужно обновить

Выполняет ввод имени

Ищет студента с таким именем в списке студентов

Если студент найден, обновляет его информацию

Если студент не найден, выводит сообщение об ошибке

Возвращает результат выполнения метода (успех/ошибка)

Класс Group:

Метод add_student():

Обновить информацию о группе в объекте студента.

Вернуть True, если операция прошла успешно, и False, если произошла ошибка.

Метод remove_student():

Удалить студента из списка студентов группы по его ID.

Обновить информацию о группе в объекте студента.

Вернуть True, если операция прошла успешно, и False, если произошла ошибка.

Метод get_students():

Получить список всех студентов, входящих в группу.

Вернуть полученный список.

Метод get_student():

Получить информацию о студенте по его ID.

Вернуть объект студента или None, если студент не найден.

Класс GroupList:

Метод add_group():

Создает новый объект класса "Group" с указанным именем и добавляет его в список групп.

Возвращает True, если операция прошла успешно, и False, если произошла ошибка.

Метод remove_group():

Удаляет группу с указанным именем из списка групп.

Возвращает True, если операция прошла успешно, и False, если произошла ошибка.

Метод get_group():

Получает информацию о группе по ее имени.

Возвращает объект класса "Group" или None, если группа не найдена.

Метод get_all_groups():

Получает список всех групп.

Возвращает список объектов класса "Group".

Класс TestSession:

Метод start_test():

Инициализирует тест, получая список вопросов из объекта Test.

Для каждого студента в списке students:

Выдает приглашение для начала тестирования.

Предоставляет студенту вопросы по одному.

Получает ответы студента на каждый вопрос.

Записывает ответы студента в объект Student.

Проверяет ответы и записывает результаты тестирования в объект Student.

Возвращает результаты тестирования.

Метод `save_test_results()`:

Получает результаты тестирования.

Записывает результаты тестирования в объект `Student`.

Возвращает результат выполнения метода (успех/ошибка).

Класс `Grader`:

представляет собой эксперта по оцениванию тестов, с методами для проверки ответов и выставления оценок.

Метод `grade_test()`:

Получает ответы студента на вопросы теста

Сравнивает полученные ответы со списком правильных ответов

Выставляет оценку за тест

Возвращает результат выполнения метода (успех/ошибка)

Метод `check_answer()`:

Сравнивает ответ студента с правильным ответом на вопрос

Возвращает результат сравнения (`True/False`)

Метод `calculate_points()`:

Вычисляет количество баллов за данную пару ответов на вопрос

Если ответы совпадают, возвращает максимальное количество баллов за вопрос, иначе - 0

Возвращает количество баллов

Класс Report:

Метод generate_report():

Получить результаты тестирования студента по его ID.

Сформировать отчет, содержащий список вопросов, ответы студента и правильные ответы.

Вернуть сформированный отчет.

Метод generate_group_report():

Получить список студентов, входящих в группу по ее ID.

Для каждого студента получить результаты тестирования и сформировать отчет.

Объединить отчеты в один общий отчет для всей группы.

Вернуть сформированный отчет.

Метод generate_summary_report():

Получить список всех студентов и их результаты тестирования.

Сформировать сводный отчет, содержащий информацию о среднем балле группы, количестве студентов, прошедших тестирование, проценте студентов, получивших оценку "хорошо" или "отлично", и т.д.

Вернуть сформированный сводный отчет.

Класс TestBank:

представляет собой хранилище тестов (список экземпляров класса Test), с методами для добавления, удаления и обновления тестов.

Метод add_test():

Добавляет новый тест в список тестов

Возвращает результат выполнения метода (успех/ошибка)

Метод remove_test():

Удаляет тест из списка тестов

Если тест не найден, выводит сообщение об ошибке

Возвращает результат выполнения метода (успех/ошибка)

Метод `update_test()`:

Обновляет информацию о тесте (например, название, дата, время и т.д.)

Если тест не найден, выводит сообщение об ошибке

Возвращает результат выполнения метода (успех/ошибка)

Класс `TestTaker`:

Метод `start_test()`:

Создание экземпляра класса `TestSession` для хранения информации о текущем тесте и результатов ответов.

Вывод на экран приглашения ввести имя студента и ввод имени.

Загрузка теста из `TestBank` с помощью метода `load_test`.

Вывод на экран текста первого вопроса и вариантов ответов из списка вопросов и ответов.

Метод `display_question()`:

Вывод на экран текста вопроса и вариантов ответов из списка вопросов и ответов.

Ввод ответа от студента.

Метод `get_student_answer()`:

Ввод ответа от студента.

Проверка корректности введенных данных с помощью метода `check_answer` из класса `Grader`.

Сохранение результата ответа в экземпляре класса `TestSession`.

Метод `check_remaining_questions()`:

Проверка наличия непройденных вопросов в текущем тесте.

Метод `generate_test_report()`:

Вывод результатов тестирования на экран с помощью метода `generate_report` из класса `Report`.

Сохранение результатов тестирования в экземпляре класса `TestResults`.

Вывод на экран сообщения о завершении теста.

Метод `run_test()`:

Вызов метода `start_test` для начала тестирования.

Вызов метода `display_question` для отображения вопроса и вариантов ответов.

Вызов метода `get_student_answer` для получения ответа студента и проверки его корректности.

Вызов метода `check_remaining_questions` для проверки наличия непройденных вопросов.

Вызов метода `generate_test_report` для вывода результатов тестирования и завершения теста.

Класс `GroupManager`:

Метод `create_group()`: создание новой группы

Метод `remove_group()`: удаление группы

Метод `assign_test_to_group()`: назначение теста на всю группу

Класс `TestGrader`:

Метод `grade_test()`: автоматическая оценка теста на основе правильности ответов

Метод `check_answer()`: проверка правильности ответа

Метод manually_grade_test(): ручная проверка теста, если это необходимо

Метод display_test_results(): отображение результатов тестирования для студента с заданным идентификатором

Класс SecurityManager:

Метод authenticate(): проверка подлинности пользователя

Метод encrypt(data): шифрование данных

Класс TestResultExporter:

Метод export_to_pdf(test_results): экспорт результатов тестирования в формат PDF

Метод export_to_excel(test_results): экспорт результатов тестирования в формат Excel

Класс StudentProgressTracker:

Метод view_test_results(): просмотр результатов тестирования для студента с заданным идентификатором

Метод view_test_dates(): просмотр дат прохождения тестов для студента с заданным идентификатором

Метод compare_results(): сравнение результатов студентов в группе с заданным именем

Метод view_group_stats(): просмотр общей статистики по группе студентов с заданным именем

Метод generate_student_report(): создание отчета о прогрессе студента с заданным идентификатором

Класс Question:

Инициализация объекта:

Передача аргументов (вопрос, варианты ответа, правильный ответ)

Создание списка объектов AnswerChoice

Возвращение объекта

Метод add_choice():

Добавление нового объекта AnswerChoice в список ответов на вопрос

Возвращение результата (успех/ошибка)

Метод remove_choice():

Поиск объекта AnswerChoice по переданному параметру choice

Удаление найденного объекта AnswerChoice из списка ответов на вопрос

Возвращение результата (успех/ошибка)

Метод update_choice():

Поиск объекта AnswerChoice по переданному параметру choice

Обновление текста ответа на новое значение new_choice_text

Возвращение результата (успех/ошибка)

Класс `LanguageSystem`:

будет иметь следующие атрибуты и методы:

Атрибуты:

`languages`: список доступных языков системы

`current_language`: текущий установленный язык системы

Методы:

инициализация объекта класса. Принимает на вход список доступных языков `languages` и язык по умолчанию `default_language`.

`set_language()`: установка языка системы. Принимает на вход язык, который нужно установить `language`. Если язык не входит в список доступных языков, метод должен вернуть `False`. Если язык установлен успешно, метод должен вернуть `True`.

`get_current_language()`: получение текущего языка системы. Метод должен вернуть текущий установленный язык.

Класс `ManuallyGrader`:

для ручной проверки тестов.

Свойства:

`test`: объект класса `Test`, который будет проверяться вручную.

Методы:

`set_test()`: устанавливает тест для ручной проверки.

`grade()`: оценивает ответ на заданный вопрос в тесте.

`get_grade()`: возвращает итоговую оценку за тест.

Алгоритм работы:

Устанавливаем тест для проверки с помощью метода `set_test`.

Поочередно выводим вопросы из теста и запрашиваем ответы на них у пользователя.

С помощью метода `grade` оцениваем ответы на каждый вопрос.

В конце работы метода `get_grade` возвращаем итоговую оценку за тест.

Класс `SetTimeLimit`:

Описание: Установка ограничений на время прохождения тестов

Методы:

`set_time_limit()` - установка временного ограничения на прохождение теста

Алгоритм:

Выдать приглашение для ввода временного ограничения

Выполнить ввод временного ограничения

Проверить, что введенное значение является целым положительным числом

Установить временное ограничение для прохождения теста

Записать результат установки ограничения в лог

Вернуть статус выполнения метода

Локальные переменные:

`time_limit` - временное ограничение на прохождение теста

Класс `CourseInfoDisplay`:

Описание:

Класс для отображения информации о прошлых и текущих курсах, а также оценок студента.

Методы:

get_student_grades(): получение оценок студента по всем курсам.

get_course_info(): получение информации о курсе (название, дата начала и окончания, преподаватель).

display_student_info(): отображение информации о студенте (ФИО, номер группы, оценки по курсам).

display_course_info(): отображение информации о курсе (название, дата начала и окончания, преподаватель, список студентов и их оценки).

display_all_courses_info(): отображение информации о всех курсах (название, дата начала и окончания, преподаватель, список студентов и их оценки).

Класс NumberRangeValidator

Ввод и контроль целого числа.

Входные данные: текст приглашения и диапазон чисел

Локальные переменные: num, ко

Алгоритм:

Выдает приглашение для ввода

Выполняет ввод

Выполняет контроль

Результат записывает в num

Повторяет ввод в случае ошибки

Если число не введено, то код ответа ко=1, иначе ко=0

Контроль:

Целое число

Входит в диапазон

Методы:

setNum – возвращает число

setKo – код ответа: 0 - норма, 1 – ввод не закончен

Класс DirectionInput.

Ввод названий всех направлений.

Входные данные: нет

Локальные переменные: парг - названия направлений в виде структуры (массив, вектор и т.д.),

kNapr – количество направлений

Алгоритм:

- Выдает приглашение для ввода направления

- Выполняет преобразование к верхнему регистру

- Записывает направление в структуру

Методы:

setNapr – возвращает направления и их количество

Класс TextFileCreator.

Создание нового текстового файла с настройками.

Входные данные: название файла, содержание первой строки файла, структура с направлениями

Локальные переменные: ко – код ответа

Алгоритм:

- Запись первой строки файла.

- Запись даты и времени создания

- Запись структуры с названиями направлений

- Запись признака конца данных(?)

Методы:

setKo – возвращает ко: ко=0 или ко=1, если файл не записан

Класс DirectionSettingsReader

Чтение названий всех направлений из файла настроек.

Входные данные: имя файла

Локальные переменные: napr - названия направлений в виде структуры (массив, вектор и т.д.),

kNapr – количество направлений, ko – код ответа

Алгоритм:

Читает направления из файла настроек в структуру napr

Методы:

setNapr – возвращает направления и их количество

setKo – возврат 0, если прочитано, 1 – если направлений нет, 2 – если файла нет

Класс DirectionInputValidator.

Ввод и контроль направления.

Входные данные: имя файла настроек

Локальные переменные: nameNapr – название направления

Алгоритм:

Читает (ReadNapr) направления из файла настроек в структуру napr

Предлагает на выбор все направления

Выбирает одно направление и записывает в nameNapr

Методы:

setNameNapr – возвращает направления

Класс `ModuleNumberInput`

Ввод и контроль номера модуля.

Входные данные: количество модулей

Локальные переменные: `nmod`, `ko`

Алгоритм:

- Выдает приглашение для ввода

- Выполняет ввод

- Выполняет контроль

- Результат записывает в `nmod`

- Повторяет ввод в случае ошибки

- Если номер не введен, то код ответа `ko=1`, иначе `ko=0`

- Контроль:

 - Целое число

 - От 1 до `n`

Методы:

`setNmod` – возвращает введенный номер модуля или ноль

`setKo` – код ответа

Класс `DataBackupManager`

Алгоритм для `backup_data()`:

- Создать резервную копию данных.

- Сохранить резервную копию на внешнем носителе или в облачном хранилище.

Вывести сообщение об успешном выполнении операции.

Методы:

backup_data(): выполняет резервное копирование данных.

Алгоритм для **restore_data():**

Запросить у пользователя идентификатор сохраненной резервной копии.

Загрузить выбранную резервную копию данных.

Восстановить данные из резервной копии.

Вывести сообщение об успешном выполнении операции.

Методы:

restore_data(): выполняет восстановление данных из резервной копии.
