

Задание

Реализация системы управления студентами и курсами

Описание

Вам предстоит разработать систему управления студентами и курсами в учебном заведении на языке Ruby. Система должна позволять добавлять студентов и курсы, регистрировать студентов на курсы и отслеживать их успеваемость. Для выполнения задания используйте классы, методы и базовые конструкции Ruby.

Технические требования

Класс Student

Атрибуты:

name (строка) — имя студента

student_id (строка) — уникальный идентификатор студента

courses (массив) — массив зарегистрированных курсов

Методы:

initialize(name, student_id) — конструктор, устанавливающий начальные значения атрибутов, courses инициализируется как пустой массив.

enroll(course) — метод для регистрации студента на курс.

to_s — метод для отображения информации о студенте в виде строки.

Класс Course

Атрибуты:

title (строка) — название курса

course_id (строка) — уникальный идентификатор курса

students (массив) — массив зарегистрированных студентов

grades (хэш) — хэш, где ключи — идентификаторы студентов, а значения — их оценки

Методы:

initialize(title, course_id) — конструктор, устанавливающий начальные значения атрибутов, students и grades инициализируются как пустые массив и хэш соответственно.

add_student(student) — метод для добавления студента на курс.

assign_grade(student_id, grade) — метод для присвоения оценки студенту.

to_s — метод для отображения информации о курсе в виде строки.

Класс University

Атрибуты:

students (массив) — коллекция студентов

courses (массив) — коллекция курсов

Методы:

initialize — конструктор, инициализирующий пустые массивы для студентов и курсов.

add_student(student) — метод для добавления студента в университет.

`add_course(course)` — метод для добавления курса в университет.
`enroll_student(student_id, course_id)` — метод для регистрации студента на курс.
`assign_grade(student_id, course_id, grade)` — метод для присвоения оценки студенту по курсу.
`list_students` — метод для отображения списка студентов.
`list_courses` — метод для отображения списка курсов.
`list_grades(course_id)` — метод для отображения списка оценок по курсу.

Указания к выполнению

Создайте классы `Student`, `Course` и `University` согласно техническим требованиям.
Напишите методы для каждого класса, реализуя описанные функции.
В основном скрипте создайте несколько студентов и курсов, добавьте их в университет.
Зарегистрируйте студентов на курсы и присвойте им оценки.
Выведите списки студентов, курсов и оценок на экран.

Пример использования

```
# Создание университета
university = University.new

# Создание студентов
student1 = Student.new("Alice", "S001")
student2 = Student.new("Bob", "S002")
student3 = Student.new("Charlie", "S003")

# Создание курсов
course1 = Course.new("Mathematics", "C001")
course2 = Course.new("Physics", "C002")

# Добавление студентов и курсов в университет
university.add_student(student1)
university.add_student(student2)
university.add_student(student3)

university.add_course(course1)
university.add_course(course2)

# Регистрация студентов на курсы
university.enroll_student("S001", "C001")
university.enroll_student("S002", "C001")
university.enroll_student("S003", "C002")

# Присвоение оценок
university.assign_grade("S001", "C001", 85)
university.assign_grade("S002", "C001", 90)
university.assign_grade("S003", "C002", 78)

# Вывод списков
puts "Students:"
university.list_students

puts "\nCourses:"
university.list_courses

puts "\nGrades for Mathematics:"
university.list_grades("C001")

puts "\nGrades for Physics:"
```

```
university.list_grades("C002")
```

Критерии оценки

1. Правильность реализации классов и методов — насколько точно реализованы методы согласно требованиям.
2. Работоспособность — корректная работа всех функций (добавление студентов и курсов, регистрация на курсы, присвоение оценок, вывод списков).
3. Чистота и структура кода — правильное использование отступов, имён переменных и методов, а также организация кода.
4. Комментарии и документация — наличие и качество комментариев, объясняющих работу методов и классов.