

ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ PYTHON (Python Beginning)

«Deadline подкрался незаметно...»
(программистский фольклор)

Урок 10. (Lesson 10) Промежуточная аттестация по пройденному материалу

Задания «Deadline по основам ООП»

1. Создайте класс *People* (*Служащий*), предназначенный для хранения данных о людях, как о сотрудниках IT-компании. Данный класс должен обеспечивать возможность для представления следующих данных:
 - фамилия, имя, отчество;
 - дата рождения;
 - возраст;
 - гражданство;
 - специальность;
 - занятость (учащийся, служащий, предприниматель, иждивенец или другое);
 - величина среднемесячного денежного дохода;
 - идентификационный номер налогоплательщика (ИНН);
 - номер свидетельства государственного пенсионного страхования;
 - номер полиса обязательного медицинского страхования и название страховой компании.

Определите конструктор для данного класса и базовый набор методов, позволяющих выполнять обработку атрибутов экземпляров класса. Перегрузите метод `__str__()`, предназначенных для строкового представления данных экземпляра класса. Сохраните разработанный класс в виде отдельного модуля.

Разработайте программы для тестирования класса *People*. Программа должна импортировать данные тестируемого модуля. Тестирующая программа должна обеспечивать возможность для сохранения введенных данных в текстовый файл и чтение данных из файла. Для вывода данных на экран воспользуйтесь кодом простого текстового редактора, рассмотренного на прошлом занятии.

2. Создайте класс *Employee* (*Сотрудник работающий по найму*), производный от класса *People*, предназначенный для представления данных о сотрудниках IT-компании. Данный класс должен содержать следующие дополнительные атрибуты:
 - уникальный номер сотрудника – ID;

- дата найма на работу в кампанию;
- должность;
- текущий проект, к выполнению которого привлечен данный сотрудник;
- величина оклада;
- величина бонусов;
- послужной список (занимаемые должности);
- список проектов, к выполнению которых был привлечен сотрудник, с указанием роли и оценки качества выполнения работы.

Определите конструктор для данного класса и базовый набор методов, позволяющих выполнить обработку атрибутов экземпляров класса. Перегрузите метод `__str__()`, предназначенных для строкового представления данных экземпляра класса. Сохраните разработанный класс в виде отдельного модуля.

Разработайте программы для тестирования класса *People*. Программа должна импортировать данные тестируемого модуля. Тестирующая программа должна обеспечивать возможность для сохранения введенных данных в текстовый файл и чтение данных из файла. Для вывода данных на экран воспользуйтесь кодом простого текстового редактора, рассмотренного на прошлом занятии.

3. Создайте класс *Job* (*Задание*), предназначенные для представления данных задания сотрудника ИТ-компании. Данный класс должен обеспечивать возможность для представления следующих данных:

- дата и время назначения задания;
- планируемая дата завершения задания;
- перечень работ, подлежащих выполнению (поэтапная детализация задания с указанием сроков выполнения работ);
- список сотрудников, участвующих в работе по выполнению данного задания;
- очередь сообщений для сотрудника компании, связанных с выполнением данного задания.

Определите конструктор для данного класса и базовый набор методов, позволяющих выполнить обработку атрибутов экземпляров класса. Перегрузите метод `__str__()`, предназначенных для строкового представления данных экземпляра класса. Сохраните разработанный класс в виде отдельного модуля.

Разработайте программы для тестирования класса *Job*. Экземпляр класса *Задание* должен быть явно связан с экземпляром класса *Сотрудник*. Программа должна импортировать данные тестируемого модуля. Тестирующая программа должна обеспечивать возможность для сохранения введенных данных в текстовый файл и чтение данных из файла. Для вывода данных на экран воспользуйтесь кодом простого текстового редактора, рассмотренного на прошлом занятии.

Рассмотрите возможность для организации очереди заданий, связанных с сотрудником. Определите возможности для системы управления заданиями – *JobManager*.

4. Определите класс *Project* (*Проект*). Данный класс должен обеспечивать возможность для представления следующих данных:
- номер проекта;
 - название проекта;
 - категория проекта (социальный, коммерческий, образовательный или другое);
 - статус проекта (планируемый, активный, завершённый);
 - результативность проекта (выполнен успешно, провален);
 - планируемый бюджет проекта;
 - фактический бюджет проекта;
 - дата и время запуска;
 - планируемая дата завершения задания;
 - фактическая дата завершения проекта;
 - перечень работ, подлежащих выполнению (поэтапная детализация задания с указанием сроков выполнения работ, ответственных за выполнение промежуточных этапов);
 - список сотрудников, участвующих в работе по выполнению данного проекта, с указанием заданий, выполнение которых им поручено;
 - очередь сообщений для сотрудника кампании, связанных с выполнением данного проекта;
 - список файлов служебной переписки, связанной с данным проектом.

Определите конструктор для данного класса и базовый набор методов, позволяющих выполнить обработку атрибут экземпляров класса. Перегрузите метод `__str__()`, предназначенных для строкового представления данных экземпляра класса. Данный метод должен обеспечивать вывод данных отчета о текущем состоянии проекта.

Сохраните разработанный класс в виде отдельного модуля.

Разработайте программы для тестирования класса *Project*. Активный *Проект* должен быть явно связан с исполнителями – экземплярами класса *Сотрудник*. Программа должна импортировать данные тестируемого модуля. Тестирующая программа должна обеспечивать возможность для сохранения введенных данных в текстовый файл и чтение данных из файла. Для вывода данных на экран воспользуйтесь кодом простого текстового редактора, рассмотренного на прошлом занятии.

Рассмотрите возможность для организации очереди заданий, связанных с выполнением работ по проекту. Определите возможности для системы управления проектами кампании – *ProjectManager*.

Кафедра «Вычислительные системы и технологии», ИРИТ, НГТУ им. Р.Е. Алексеева

Изучите современные программные пакеты, предназначенные для управления проектами, такие как MS Project и имеющиеся его бесплатные аналоги, например, ADVANTA.

Не опубликованная версия, замечания и предложения направляйте на кафедру