

Оглавление

Введение.....	1
Цель проекта	1
Задачи проекта.....	1
Организация работ	2
Требования к реализации проекта	3
Интерфейс	3
Структура базы данных.....	3
Операции над базой данных	3
Состав отчетов	4
Структура приложения	4
Организация кода.....	5
Документация	5

Введение

Настоящий документ содержит постановку задачи для самостоятельной работы слушателей курса «Проектный семинар «Python в науке о данных» и **описание минимальных требований к ее результатам. Выполнение данных требований необходимо для получения оценки «удовлетворительно» с минимальным количеством баллов.** Оценка за самостоятельную работу является экспертной оценкой преподавателем качества выполнения задания. Оценка проставляется по десятибалльной шкале.

Цель проекта

Продемонстрировать умение использовать язык Python 3.X (реализация CPython в составе дистрибутива Anaconda, версии текущей на начало обучения), его стандартной библиотеки (**кроме SQLite**), а также библиотек NumPy, Pandas и Matplotlib (версий, текущих на начало обучения) для разработки специализированных информационно–аналитических приложений.

Задачи проекта

Основной задачей проекта является разработка специализированного информационно–аналитического приложения средствами языка Python (CPython) 3.X, его стандартной библиотеки, библиотек NumPy, Pandas и Matplotlib (версий, текущих на начало обучения), а также документации к ней — руководства пользователя и руководства

разработчика. Обращаем внимание на то, что использование других библиотек, помимо указанных, не разрешено.

Организация работ

Реализация проекта осуществляется в течение двух (третий и четвертый) модулей в группах (бригадах) по три – четыре человека. Один из участников группы выбирается Директором проекта, основной задачей которого является организация взаимодействия бригады с преподавателем. С этой целью он предоставляет преподавателю адрес своей электронной почты, номер своего мобильного телефона и, при наличии, идентификатор в skype.

Все работы по реализации проекта распределяются между членами бригады в равных долях исключительно по обоюдному соглашению между ними. Вклад каждого оценивается преподавателем индивидуально. Таким образом, оценки членов бригады за проект могут быть разными.

Все взаимодействие со студентами осуществляется, либо лично, на занятиях, либо удаленно на платформе **slack**. С этой целью Директорам проектов будут высланы приглашения для регистрации в **slack**.

После прохождения процедуры регистрации Директора заполняют свои профили, внося в них: свои имя, фамилию и отчество (добавить номер группы), номер мобильного телефона, адрес электронной почты и, при наличии, skype. Также каждый Директор выкладывает в профиль свою фотографию. В поле «What I do» Директор указывает название проекта своей бригады и номер группы.

Бригады каждой группы объединены в каналы. Номер канала соответствует номеру группы. Взаимодействие с преподавателем осуществляется в каналах групп. Все важные общие сообщения (в частности, ведомости), а также учебная литература размещаются преподавателем в канале **#general**. Директор обязан не менее двух раз в неделю просматривать этот канал. Канал **#random** используется для свободного взаимодействия между студентами и с преподавателем. Преподаватели осуществляют непрерывное консультирование студентов по вопросам, связанным с освоением курса. Свои вопросы, промежуточные и окончательные результаты работы Директор направляет в **slack** в канал своей группы. Свои комментарии преподаватель направляет в канал группы или лично Директору проекта и при необходимости копирует в канал **#general**. Обращаем внимание на то, что использовать электронную почту и мобильный телефон при обращении к преподавателю разрешено только в крайнем случае, если возникает нештатная ситуация или связаться с преподавателем лично или в **slack** не удастся в течение, как минимум, недели.

Предметная область, а также данные для проектов (содержание баз данных) готовятся/выбираются студентами самостоятельно по согласованию с преподавателем.

В ходе реализации проекта каждая бригада может дважды обратиться к преподавателю за подробной консультацией по проекту. Для консультации Директор в **slack** предоставляет преподавателю текущую версию кода, а также накопившиеся вопросы

по реализации проекта. Каждый вопрос должен содержать вариант ответа, который Директор проекта считает правильным — «Правильно ли мы понимаем, что ...?», «Может ли данная проблема быть решена таким образом ...?» и т.д. В свою очередь преподаватель комментирует предоставленный код и, по возможности, дает ответы на вопросы также в **slack**.

Финальная версия информационно-аналитического приложения, «Руководства пользователя» и «Руководства разработчика» предоставляется бригадой преподавателю в **slack** не позднее, чем за 10 дней до начала сессии. Настоятельно рекомендуем до этого тщательно протестировать приложение на различных компьютерах.

Проверка преподавателем работоспособности созданного приложения, а также его соответствия заявленному в «Руководстве пользователя» функционалу осуществляется из командной строки, а не в среде разработки, запуском на счет основного скрипта. Это необходимо учитывать при проектировании и тестировании созданного ПО.

При необходимости преподаватель может назначить защиту проекта для всей бригады. В ходе защиты участники бригады должны будут ответить на вопросы об организации кода разработанного приложения.

Требования к реализации проекта

Интерфейс

Участники бригад разрабатывают информационно-аналитическое приложение с графическим интерфейсом **для платформы Widows**, реализованным на платформе Tcl/Tk. Графический интерфейс является максимально дружелюбным и интуитивно понятным специалисту в выбранной функциональной области, не содержит специальной (математической и ИТ) терминологии, полностью русскоязычный.

Структура базы данных

Все отношения (таблицы) базы данных должны быть приведены к третьей нормальной форме. Полученные отношения должны использоваться как справочники при построении отчетов. Схемы отношений должны включать как числовые, так и качественные атрибуты. Т.е. столбцы таблиц, образующих базу данных, должны содержать, как столбцы с качественными данными, так и с числовыми. **База данных должна состоять как минимум из двух справочников.**

Операции над базой данных

Функционал информационно-аналитического приложения должен включать в себя два вида операций над базой данных:

- ручное добавление/удаление сущностей (объектов) на основе существующих справочников и ввода значений атрибутов;
- ручная модификация существующих справочников;
- сохранение базы данных в двоичном формате, чтение базы данных из двоичного формата.

При подготовке текстовых отчетов необходимо использовать инструменты объединения таблиц из библиотеки Pandas.

Состав отчетов

Функционал информационно-аналитического приложения должен включать в себя следующие виды отчетов:

- простой текстовый отчет, полученный за счет использования операций проекции и сокращения, т.е. таблица, полученная вычеркиванием части строк и столбцов;
- текстовый статистический отчет по любому набору атрибутов (набор основных описательных статистик);
- текстовый отчет «сводная таблица» для любой пары качественных атрибутов с выбором метода агрегации;
- графический отчет «кластеризованная столбчатая диаграмма» для пары «качественный атрибут — качественный атрибут»;
- графический отчет «категоризированная гистограмма» для пары «количественный атрибут— качественный атрибут»;
- графический отчет «категоризированная диаграмма Бокса-Вискера» для пары «количественный атрибут— качественный атрибут»;
- графический отчет «категоризированная диаграмма рассеивания» для двух количественных атрибутов и одного качественного атрибута.

Для всех текстовых отчетов предусмотрена опция вывода в файл. Все графические отчеты могут быть экспортированы в графические файлы (формат по выбору).

Структура приложения

Информационно–аналитическое приложение размещается в стандартной структуре каталогов:

Work <- основной каталог.

Data — содержит базу данных.

Graphics — содержит копии графических отчетов.

Library — содержит библиотеку стандартных (универсальных) функций, разработанных бригадой, которые могут использоваться для создания других приложений, например, функции чтения файлов.

Notes — содержит документацию, в нем размещается Руководства пользователя и разработчика.

Output — содержит копии текстовых отчетов.

Scripts — содержит специализированный модуль и файл с определением параметров настройки приложения.

Организация кода

Код разработан в соответствии с требованиями «Python Enhancement Proposals (PEP) 8 -- Style Guide for Python Code» и «PEP 257 -- Docstring Conventions» (см. <https://www.python.org/dev/peps/>)

Параметры настройки приложения (цвета, размеры окон и шрифты, а также пути к базе, отчетам и т.д.) размещены в текстовом файле, который импортируется в начале работы приложения. Код содержит интерактивный блок настройки основных параметров, значения которых опционально сохраняются в указанный файл параметров настройки.

Все функции и классы, созданные в ходе реализации проекта, объединяются в два модуля, импортируемых в начале работы приложения. Один содержит стандартные (универсальные) функции и классы, разработанные бригадой, которые могут использоваться для создания других приложений. Он размещается в каталоге **library**. Второй модуль содержит специализированные функции и классы, разработанных бригадой, необходимые для работы данного приложения. Он размещается в каталоге **scripts** вместе с главным скриптом.

Все функции и классы содержат строки документации, в которых приведены: описание функции/класса, описание входных параметров, описание возвращаемого объекта, автор кода.

Код содержит достаточное количество комментариев (помимо строк документации) для понимания его структуры.

Документация

Руководст во пользоват еля содержит: описание назначения данного информационно–аналитического приложения, требования к характеристикам компьютера и операционной системе, инструкцию по установке приложения, инструкцию по запуску и настройке приложения, инструкцию по использованию функционала приложения, в частности, указано, где находятся копии отчетов. *Руководст во разработ чика* содержит: требования к характеристикам компьютера и операционной системе, указание версий языка и библиотек, использованных при разработке приложения, описание архитектуры приложения (из каких частей состоит и как они между собой связаны), описание структуры каталогов, листинг основного скрипта и всех модулей.