

Оглавление

Введение	1
Цель проекта	1
Задачи проекта	1
Критерии оценивания	2
Организация работ	3
Требования к реализации проекта	4
Интерфейс	4
Структура базы данных	5
Операции над базой данных	5
Состав отчетов	5
Структура приложения	6
Организация кода	6
Документация	7

Введение

Настоящий документ содержит постановку задачи для самостоятельной работы слушателей курса «Проектный семинар «Python в науке о данных» и описание требований к ее результатам. 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 учебного курса), его стандартной библиотеки (кроме SQLite), а также библиотек NumPy, Pandas и Matplotlib (версий, входящих в состав дистрибутива Anaconda актуального для даты начала работы проектного семинара) для разработки специализированных информационно–аналитических приложений.

Задачи проекта

Основной задачей проекта является разработка для платформы MS Windows 10 специализированного информационно–аналитического приложения средствами языка Python (CPython) 3.X, его стандартной библиотеки (кроме SQLite), библиотек NumPy, Pandas и Matplotlib (версий, входящих в состав дистрибутива Anaconda актуального для даты начала работы проектного семинара), а также документации к нему — руководства пользователя и руководства разработчика. Обращаем внимание на то, что использование других библиотек, помимо указанных, не разрешено. Приложение должно запускаться из командной строки: `python <имя главного модуля>.py`. Возможно использование аргументов командной строки. Приложение

сопровождается двумя документами: «Руководство пользователя» (инструкция по установке и использованию, которая используется при проверке приложения) и «Руководство разработчика» (описание системных требований, названий и версий библиотек, а также структуры приложения).

Предметная область информационно-аналитических приложений, а также данные для проектов (содержание баз данных) готовятся/выбираются студентами самостоятельно по согласованию с преподавателем. Данные в БД должны быть реальными или похожими на реальные. Количество данных должно быть достаточным для построения статистических отчетов – 100—200 объектов. **В рамках одной группы предметные области не должны пересекаться.** Например, можно создавать приложение для врача школы, директора универсама, оператора авиа порта, продавца косметики или модной одежды и т.д.

Критерии оценивания

Отличная реализация требований задания для самостоятельной работы приводит к оценке «отлично» с восемью баллами. Получение большого количества баллов возможно только при признании реализации *выдающейся*, т.е. обладающей существенными дополнительными возможностями, выходящими за рамки стандартного задания.

- Работоспособность приложения. Приложение запускается из командной строки: python <имя главного модуля>.py. При запуске и использовании функционала, указанного в «Руководстве пользователя» не должно возникать ошибок и не должно выводиться в консоль сообщений, связанных с некорректной работой приложения. Возможности языка Python и специально подобранные требования к разрабатываемому продукту позволяют с уверенностью утверждать, что следование указаниям задания для самостоятельной работы гарантирует работоспособность создаваемого при выполнении задания приложения на любом компьютере, на котором установлена общая для всех студентов версия дистрибутива Python — Anaconda, при отсутствии ошибок разработчиков данного приложения. Таким образом, если разработчик корректно создал код приложения оно должно работать, как на его компьютере, так и на компьютере другого пользователя, в частности, преподавателя, при условии выполнения указанных выше требований. Отсутствие возможности приступить к эксплуатации приложение может свидетельствовать только, либо о наличии ошибок в коде, либо о нарушении требований задания, в частности, использовании специфических особенностей компьютера разработчика. Отсутствие возможности приступить к работе с приложением (критическая ошибка¹ при запуске) или прекращение работы приложения по причине возникновения критической ошибки в ходе его эксплуатации автоматически приводит к оценке **ноль баллов по десятибалльной системе**. Наличие не критических ошибок приводит к адекватному снижению балла за приложение.
- Структура приложения. Полнота и качество реализации задания для самостоятельной работы, в частности, наличие не менее двух справочников и их соответствие третьей

¹ Критической ошибкой называется ошибка работы кода приложения, которая делает невозможным его дальнейшую эксплуатацию.

нормальной форме, наличие возможности работы со справочниками, образующими базу данных, количество отчетов и качество их реализации, наличие дополнительных опций, согласованных с преподавателем для получения оценки девять или десять. Приложение должно соответствовать всем требованиям, указанным в данном документе. Отсутствие тех или иных требуемых элементов приводит к адекватному снижению балла за приложение. Полное несоответствие требованиям приводит к оценке **ноль баллов по десятибалльной системе**.

- Интерфейс. Качество реализации и удобство работы с интерфейсом всех частей приложения — минимально возможное наличие ручного ввода данных, интуитивная понятность и удобство размещения элементов графического интерфейса (виджетов), возможность конфигурирования интерфейса (настройки цветов и шрифтов, а также управления окнами с помощью мыши). Минимально допустимой реализацией настройки интерфейса является использование конфигурационных файлов. Приветствуется наличие графического интерфейса для выполнения настроек. Отсутствие возможности настраивать интерфейс является нарушением требований и приводит к оценке **ноль баллов по десятибалльной системе**.
- Реализация кода. Код должен быть 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/>). Оценка качества кода в IDE Spyder должна быть не ниже 5 баллов. Более низкая оценка приводит к оценке **ноль баллов по десятибалльной системе**. Более высокий балл является аргументом для повышения оценки. Код должен содержать много комментариев, поясняющих назначение всех его частей. Количество и качество комментариев влияет на оценку.
- Документация к приложению. Оценивается качество реализации документации — «Руководства пользователя» и «Руководства разработчика». Документы должны иметь пронумерованные страницы и разбиты на разделы. В документах должно присутствовать оглавление, созданное средствами MS Word. При реализации в формате Adobe Acrobat, оглавление должно обеспечивать навигацию по документу. Технические требования: шрифт Calibri Light 12, межстрочный интервал 1.15, все таблицы и рисунки (скриншоты) пронумерованы и имеют название, каждый документ имеет титульный лист с указанием названия учебного заведения, учебной программы, учебной группы, номера и состава бригады (приведен телефон и адрес Директора).
- Плагат. Совпадение пятидесяти и более процентов строк кода в приложениях двух бригад считается плагиатом и бригада, совершившая плагиат, получает оценку **ноль баллов по десятибалльной системе**. Если ни одна из бригад не признает плагиат, то обе бригады получают оценку **ноль баллов**. В связи с этим рекомендуется выкладывать в Telegram группу запароленные архивы с приложениями.

Организация работ

Реализация проекта осуществляется в течение двух (третий и четвертый) модулей в группах (бригадах) по два – три 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 и номер своего мобильного телефона.

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 Telegram. Для каждой учебной группы создается отдельная группа в Telegram. Все создаваемые в ходе работы документы и приложения выкладываются только в Telegram –группу учебной группы Директорами бригад. Использование личных сообщений не разрешается, личные чаты сразу удаляются. Вопросы преподавателям в Telegram-группе могут задавать все ее участники. Каждый вопрос должен содержать вариант ответа, который задающий вопрос считает правильным — «Правильно ли мы понимаем, что ...?», «Может ли данная проблема быть решена таким образом ...?» и т.д.

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 Telegram не удастся в течение, как минимум, недели.

Финальная версия информационно-аналитического приложения, «Руководства пользователя» и «Руководства разработчика» выкладывается в Telegram-группу Директором бригады 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.

Финальная версия, а также промежуточные версии, предоставляются в виде архивов каталогов «work», содержащих все компоненты проекта (подробное описание дано ниже). Название архива - <Группа (три символа)> _ <Бригада (два символа)> _ <Версия (три символа)>.zip. Например, 191_01_003.zip. Архивы должны иметь постоянный на весь срок обучения пароль, который Директор сообщает преподавателю в личном сообщении в Telegram. Менять структуру каталогов, а также формат названий не разрешается.

Требования к реализации проекта

Интерфейс

Участники бригад разрабатывают информационно-аналитическое приложение с графическим интерфейсом для платформы Windows 10, 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. Настройка интерфейса хранится в

конфигурационном текстовом файле с расширением .ini и может (но не обязательно) меняться с помощью специального интерфейса в приложении.

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. Справочники хранятся на диске в каталоге work/data и загружаются при старте приложения. Все прочие таблицы формируются из указанных справочников.

Операции над базой данных

Для каждого справочника приложение должно содержать отдельный графический интерфейс для управления справочником. Функционал информационно-аналитического приложения должен включать в себя следующие виды операций над базой данных:

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

Состав отчетов

Приложение должно предоставлять возможность формировать ряд информационно-аналитических отчетов. Для формирования отчетов каждого типа должен быть сформирован отдельный графический интерфейс. Пользователь должен иметь возможность просмотра результата построения отчетов непосредственно в приложении и сохранить его при необходимости.

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. При подготовке текстовых отчетов необходимо использовать инструменты объединения таблиц из библиотеки Pandas.;
- текстовый статистический отчет по любому набору атрибутов (набор основных описательных статистик, отчет включает в себя:
 - для качественных переменных (например, пол, место проживания и т.д.) таблицу частот — первый столбец содержит уровни (значения) переменной, второй столбец содержит частоты, т.е. количество объектов с данным уровнем фактора, третий столбец — процент количества указанных объектов от их общего числа
 - для количественных переменных — основные статистики, т. е. максимум и минимум, арифметическое среднее, выборочную дисперсию и стандартное отклонение, отчет оформлен в виде таблицы — первый столбец список переменных, далее в каждом столбце значение статистики, значение статистик подсчитывается с помощью методов класса pandas.DataFrame;

- текстовый отчет «сводная таблица» для любой пары качественных атрибутов с выбором метода агрегации, сводная таблица строится с помощью функции `pandas.pivot_table()`;
- графический отчет «кластеризованная столбчатая диаграмма» для пары «качественный атрибут — качественный атрибут», следует использовать `matplotlib.pyplot.bar()`;
- графический отчет «категоризированная гистограмма» для пары «количественный атрибут—качественный атрибут», следует использовать `matplotlib.pyplot.hist()`;
- графический отчет «категоризированная диаграмма Бокса-Вискера» для пары «количественный атрибут—качественный атрибут», следует использовать `matplotlib.pyplot.boxplot()`;
- графический отчет «категоризированная диаграмма рассеивания» для двух количественных атрибутов и одного качественного атрибута, следует использовать `matplotlib.pyplot.scatter()`.

Для всех текстовых отчетов должна быть 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/>)

При разработке приложения не разрешается использовать объектно-ориентированное программирование. Эта тема будет изучаться позднее в других курсах. Следует использовать функциональный подход — разрабатывать функции и библиотеки функций, а не классы.

Параметры настройки приложения (цвета, размеры окон и шрифты, а также пути к базе, отчетам и т.д.) размещены в конфигурационном текстовом файле с расширением `.ini`, который импортируется в начале работы приложения. Код может содержать 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 пользует 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.