

Темы домашних заданий
по дисциплине «Разработка приложений на языке Python»
для направления 09.04.01 – «Информатика и вычислительная техника».
Программа «DevOps-инженерия интеллектуальных корпоративных систем»

Требования к разрабатываемому ПО:

1. Программное обеспечение должно быть написано на языке программирования Python. Допускается изменение языка программирования по согласованию с преподавателем.
2. Все модули должны работать на локальном сервере без выхода в сеть Интернет (кроме заранее согласованных с преподавателем вариантов).
3. Допускается использование любых методов, функций, классов и т.д., которые не противоречат настоящему заданию.
4. Не допускается использование разработанных другими авторами блоков кода.
5. При разработке допускается использование любого вспомогательного ПО при условии лицензионной чистоты этого ПО.
6. Все данные, если это не указано отдельно, должны храниться в БД (с использованием, например, СУБД MySQL).
7. Должно быть разработано собственное клиентское ПО (не допускается использование стороннего клиентского ПО, например, браузера).
8. Все логика работы ПО должна осуществляться на стороне сервера (если обратное явно не указано в задании).

Требования к оформлению:

1. Отчет по домашнему заданию должны содержать следующие разделы:
 - титульный лист;
 - задание;
 - структуру системы с описанием;
 - листинг программного кода;
 - структуру БД и отдельных таблиц (если используется БД);
 - пример выполнения программы (скриншоты);
 - инструкцию по эксплуатации ПО;
 - выводы;
 - список используемой литературы.
2. Отчет и архив проекта должен быть загружен в LMS Moodle.

Допускается разработка ДЗ по индивидуальному заданию (студент предлагает свой вариант). Обязательно должно быть согласовано с преподавателем. Решение задач в сфере DevOps добавляет +4 балла. Допускается выполнение групповых проектов. При этом в отчете должна быть описана роль каждого участника проекта.

Тему задания (типового, или индивидуального) необходимо сообщить преподавателю (отправить в личный чат MS Teams).

Задания

Вариант 1.

Необходимо разработать ПО, позволяющее загружать изображения формата JPG в отдельную папку на сервере. Перед загрузкой на сервер необходимо уменьшать изображение до заданного размера. Если изображение меньше заданного размера, выдавать сообщение об ошибке. Необходимо учесть, что имена файлов с изображениями могут совпадать.

Вариант 2.

Необходимо разработать систему мониторинга заданного пользователем сайта в поисковых системах Yandex, Rambler, Google по указанному пользователем запросу. ПО должно выводить список сайтов с 1-ого места по место указанного сайта (но не далее 50 места).

Вариант 3.

Разработать органайзер на текущий месяц. На главном экране должен выводиться календарь на текущий месяц. При щелчке на календарном дне, должна выводиться информация по этому дню. В том случае, если пользователь авторизован, то должна быть обеспечена возможность редактирования и добавления информации по указанному дню.

Вариант 4.

Разработать ПО, которое меняет заданную текстовую строку на другую строку во всех файлах в указанной директории и ее поддиректориях. Вся исходная информация должна задаваться в графическом интерфейсе клиентского ПО. Лог работы должен храниться в БД.

Вариант 5.

Разработать ПО, которое позволяет загружать на сервер файл по протоколу FTP. Предусмотреть возможность совпадения имен файлов. При правильной авторизации выводить список всех доступных файлов с их размерами и предоставлять возможность скачать выбранный файл. Лог работы должен храниться в БД.

Вариант 6.

Разработать скрипт, который считывает текстовый файл с сервера и выводит содержимое этого файла на экран. При этом, если файл содержит код Python, то синтаксис языка должен подсвечиваться аналогично подсветке IDE Visual Studio (или PyCharm). Не допускается использование функций для подсветки текста, разработанных другими авторами.

Вариант 7.

Разработать ПО гостевые книги, которое позволяют просматривать ранее добавленные записи. При правильной авторизации должна быть предусмотрена возможность добавления, удаления и редактирования записей. При добавлении записи предусмотреть возможность добавления смайла (не менее 5 видов). При авторизации администратором должна быть предусмотрена возможность добавлять, удалять и редактировать учетные записи (логин, пароль и примечание) пользователей. Лог работы должен храниться в БД.

Вариант 8.

Разработать два программных модуля. Первый модуль генерирует данные по вершинам дерева в БД. Каждая вершина содержит номера дочерних узлов (максимум три дочерних узла). Наличие, или отсутствие дочернего узла должно определяться по случайному алгоритму. Второй модуль должен отображать сгенерированную информацию в виде дерева (горизонтального, или вертикального). Допускается совмещение модулей в единое ПО.

Вариант 9.

Разработать ПО, которое автоматически определяет кодировку указанного текстового файла (файл находится на сервере). Файл может содержать текст на русском языке и цифры. Не допускается использование готовых функций определения кодировки. Лог работы должен храниться в БД.

Вариант 10.

Разработать ПО, которое заносит информацию в БД из указанного RSS-канала. Должна добавляться следующая информация: заголовок, ссылка, описание, время. Должны быть предусмотрены следующие фильтры: по дате добавления, по размеру описания, по ключевым словам (не более 3-х с логическими связками И, ИЛИ, НЕ). Должна быть предусмотрена возможность вывода информации по указанному каналу из БД на экран.

Вариант 11.

Серверное ПО, которое собирает статистику обращений к нему. При этом в БД должна сохраняться следующая информация: браузер + ОС, IP адрес, хост, страницу-реферер, время визита. Необходимо обеспечить возможность вывода вышеуказанной информации на экран за указанный временной промежуток.

Вариант 12.

Разработать ПО, которое бы позволяло скачивать файлы с сервера при этом предварительно архивируя их (любыми средствами сервера). При скачивании на экран должны выводиться истинный размер файла, размер сжатого файла и коэффициент сжатия. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 13.

Разработать ПО, которое позволяет осуществлять поиск указанной текстовой строки в текстовых файлах в указанной директории и ее поддиректориях. В результате поиска на экран должны выводиться пути, имена файлов и номера строк в которых обнаружена заданная строка. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 14.

Разработать ПО, которое выводит на экран информацию о владельце доменного имени и его контактных данных. Лог запросов (IP, имя домена, время запроса) должен сохраняться в БД. Необходимо использовать сайты регистраторов доменов без использования API. Все действия должны осуществляться через графический UI.

Вариант 15.

Разработать клиент-серверное ПО, которое перекодирует русский текст из указанного файла на сервере в транслит. Результат перекодировки записываются в новый файл на сервере. Лог запросов (IP, имя файла, время запроса) должен сохраняться в БД. Перекодировка должна производиться на стороне сервера. Все действия пользователя должны осуществляться через графический UI.

Вариант 16.

Необходимо создать ПО, который бы показывал список пользователей, которые её посещали и сколько раз. Посетитель должен иметь возможность:

- а) ввести свое имя, чтобы в этот список попасть.
- б) искать другого человека, зная его точное имя.
- в) нажать на ссылку в списке, чтобы удалить запись из него (после ввода пароля).

Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 17.

Разработать ПО, создающие точечную диаграмму (в графическом виде) по следующим исходным параметрам: width, height - ширина и высота изображения (задаются через форму).

data.dat - файл, содержащий данные (на каждой строке два числа, разделенные точкой с запятой: первое число - величина, откладываемая на горизонтальной оси, второе число откладывается на вертикальной оси). Редактирование файла с исходными данными должно быть предусмотрено в клиентском ПО. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 18.

Разработать ПО, которое при успешной авторизации (логин/пароль находятся в БД) выводит вертикальное дерево файлов (имена и размеры) из указанной серверной директории и ее поддиректорий. Уровень вложения не менее 5. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 19.

Необходимо создать ПО, которое по числовым данным из файла на сервере (число-название-цвет в формате RGB) построит круговую диаграмму, каждый сектор которой будет соответствовать процентному отношению указанного числа к общей сумме. Все сектора должны быть раскрашены в разный цвет. Под диаграммой следует вывести легенду. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 20.

Разработать ПО для игры Судoku. Варианты игры (не менее 10) должны быть предварительно заданы в БД. Уровень сложности (кол-во открытых цифр) задается перед началом игры. Информация должны представляться в графическом виде. Лог работы должен храниться в БД.

Вариант 21.

Разработать скрипт, который принимает отзыв из формы (имя, содержание, дату) и отправлял введенные данные на указанный адрес электронной почты. Предусмотреть систему защиты с использованием ручного ввода содержимого графического изображения. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 22.

Разработать ПО для контроля знаний студентов. ПО должно позволять администратору (после авторизации) добавлять, удалять, редактировать контрольные вопросы (вопрос, варианты ответов и максимальный балл). После авторизации пользователь проходит тесты и получает оценку. Все ответы должны сохраняться в БД.

Вариант 23.

Разработать ПО фотоальбома, которое должно позволять пользователю (после авторизации) добавлять, удалять и изменять подписи к фотографиям, загруженным этим пользователем, и просматривать фотографии других пользователей. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 24.

Разработать файловый менеджер. ПО должно отображать информацию о файлах и директориях, позволять удалять директории и файлы, создавать директории и загружать файлы. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 25.

Разработать ПО, реализующее доску объявлений. Оно должно позволять пользователю (после авторизации) выбирать раздел, добавлять, удалять и редактировать объявление. ПО должно отображать дату добавления, количество просмотров и автора объявления. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 26.

Разработать ПО, реализующее игру «Крестик-нолики». Игровое поле 3*3. Вся логика работы ПО должна быть реализована на стороне сервера. В качестве клиента может использоваться браузер. История ходов должна храниться в БД.

Вариант 27.

Разработать ПО, которое позволяет работать с данными в таблице БД. Таблица должна иметь фиксированный размер. Необходимо обеспечить выполнение следующих операций: получение всех данных из таблицы, вставка новой строки, изменение существующей строки, удаление строки. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 28.

Разработать ПО, которое позволяет просматривать список всех БД в СУБД, список всех таблиц/триггеров/процедур/функций, а также удалять указанную БД/таблицу/триггер/процедуру/функцию. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 29.

Разработать ПО, которое позволяет получать информацию о новостях из RSS-ленты (находится на удаленном ресурсе, который выбирает студент). Необходимо получать: заголовок, аннотацию, дата/время, ссылку на полную новость. Полученная информация должна сохраняться в БД. Должны быть предусмотрены следующие фильтры: диапазон дат, список запрещенных слов, список необходимых слов. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 30.

Составить программу, которая содержит текущую информацию о книгах в библиотеке. Сведения о книгах содержат: номер УДК, фамилию и инициалы автора, название, год издания, количество экземпляров в библиотеке. Программа должна обеспечить: начальное формирование данных обо всех книгах; добавление данных о книгах, поступающих в библиотеку; изменение данных при вводе информации о том, что пользователь берет или возвращает книгу; выдачу данных о наличии книг в библиотеке. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 31.

Составить программу, которая моделирует заполнение гибкого диска. В процессе работы файлы могут записываться на диск и удаляться с него. Файлы могут иметь произвольную длину. Если при удалении образовался свободный участок, то вновь записываемый файл помещается на этом свободном участке, либо, если он не помещается на этом участке, то его следует разместить после последнего записанного файла. Если файл превосходит длину самого большого участка, выдается аварийное сообщение. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.

Вариант 32.

Составить программу, которая содержит информацию о квартирах, содержащихся в базе данных бюро обмена квартир. Сведения о каждой квартире содержат: количество комнат; общую площадь; этаж; адрес. Программа должна обеспечить: формирование картотеки; ввод заявки на обмен; поиск подходящего варианта (при равенстве количества комнат и этажа и различии площадей в пределах 10% выводится соответствующая карточка, а сами сведения удаляются из списка, в противном случае поступившая заявка включается в список); вывод всего списка. Все действия должны осуществляться через графический UI. Лог работы должен храниться в БД.