Санкт-Петербургский Национальный Исследовательский Университет Информационных технологий, механики и оптики

**Лабораторная работа 1**

**Реализация программной модели инфокоммуникационной системы**

Выполнил: Касьяненко

Вера Михайловна

Группа № K3121

Проверила: Казанова

Полина Петровна

Санкт-Петербург

2022

**Цель работы:**

Создать программное обеспечение системы обработки данных: «Программа для контроля собственных денежных средств».

**Задачи:**

* Провести анализ предметной области и требований.
* Придумать алгоритмы для решения конкретных задач.
* Написать программу для данных алгоритмов на языке Python.
* Получить требуемый результат, убедившись в работоспособности программы.

**Задание на выполнение домашнего задания:**

Создать программное обеспечение системы обработки данных: «Программа для контроля собственных денежных средств» со следующими функциями:

1. Добавлять продукт в коллекцию (тип коллекции на ваш выбор).
2. Просматривать все записанное в программу.
3. Просматривать покупки по дате и категории.
4. Распределять их по стоимости от минимальной к максимальной или наоборот.
5. Удалять требуемые записи и выходить из программы.

**Ход работы:**

Перед началом создания алгоритма необходимо провести анализ предметной области и требований. Наиболее похожим программным обеспечением являются приложения банков, которые позволяют контролировать расходы. Однако в отличие от приложений банков, в программном обеспечении, которое предстоит создать, пользователь сам вводит данные о покупках. Подобные приложения чаще всего используется обычными пользователями, которые не обладают навыками программиста, поэтому необходимо создать программу, которая будет понятна и проста в использовании. Также программа должна выполнять поставленные перед ней задачи, например создание и просмотр отчетов о покупках, совершенных пользователем.

Для того, чтобы программа функционировала, необходимо создать файлы с базами данных, которые будут использоваться в дальнейшем:

1. Файл с именем пользователя.
2. Файл с базой продуктов, их стоимостью и категорией.
3. Файл записями о покупках.
4. Файл с записью о текущем балансе.

Вся программа состоит из функций, которые вызываются при необходимости. Рассмотрим их:

1. Функция приветствия пользователя. Программа считывает имя из файла в переменную (рисунок 1) и выдает его при начале работы программы   
   (рисунок 2). Если имени в файле не обнаружено, то программа печатает приветствие на экран без имени (рисунок 3).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Функция, показывающая приветствие



Рисунок 2 – Приветствие с именем



Рисунок 3 – Приветствие без имени

Более того, в программе предусмотрено изменение имени. На рисунке 4 наглядно представлен алгоритм изменения имени, а также функция на рисунке 5 и ее демонстрация на рисунке 6.

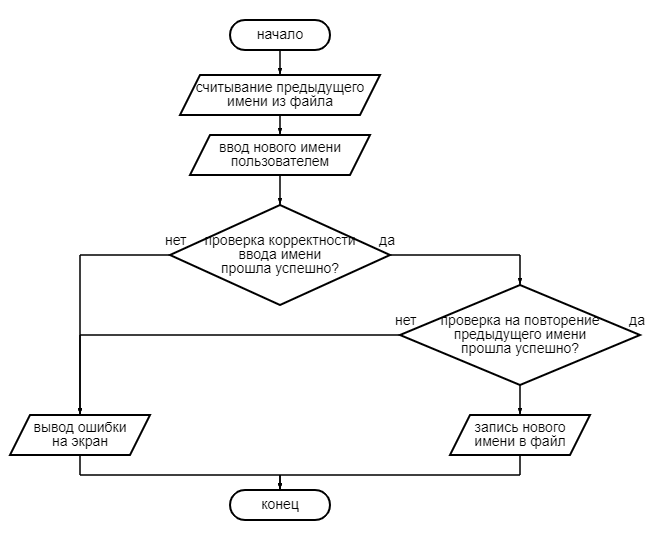


Рисунок 4 – Алгоритм изменения имени

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 5 – Функция изменения имени

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 6 – Демонстрация изменения имени

При вызове функции изменения имени введенное имя записывается в переменную, после чего проверяется на корректность ввода и повторение предыдущего имени при помощи условных операторов, если проверки пройдены, то новое имя записывается в файл, иначе выводится сообщение об ошибке (рисунок 7).

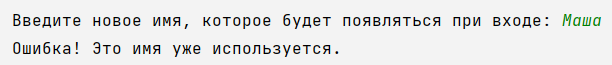


Рисунок 7 – Ошибка, если имя совпадает с предыдущим

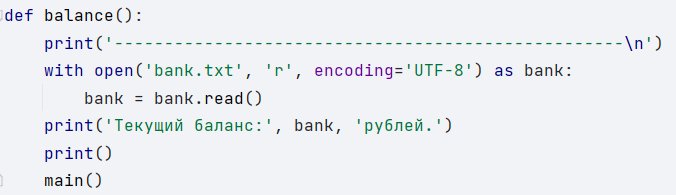
1. Функция просмотра (рисунок 8) и изменения баланса (рисунок 9). Для лучшего контроля расходов была написана функция, которая считывает сумму, добавленную пользователем в приложение, и работает с ней. Данные о балансе так же, как и имя, хранятся в файле. При работе с приложением пользователь может просмотреть или изменить свой баланс. 

Рисунок 8 – Функция просмотра баланса

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 9 – Функция изменения баланса

При просмотре баланса производится считывание из файла в переменную и печать этой фразы на экран с этой переменной (рисунок 10). При изменении баланса так же происходит считывание текущей суммы из файла. Далее пользователь вводит новую сумму, которая проверяется на правильность ввода (например сумма не может быть отрицательной или пустой) при помощи поэлементного сравнения с недопустимыми символами, а также проверяется на повторение предыдущей суммы с помощью условного оператора, после чего сумма записывается в файл, используется как новая, а также выводится на экран (рисунок 11). В противном случае выводится сообщение об ошибке.



Рисунок 10 – Вывод баланса

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 11 – Изменение баланса

1. Функция просмотра и изменения базы продуктов. Для последующей работы пользователь добавляет продукты в базу, указывая их стоимость и категорию (рисунок 12). При вводе данных происходит проверка на присутствие данного продукта в базе (рисунок 13). Во избежание ошибок с регистром, веденные данные обрабатываются методом capitalize().

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 12 – Функция добавления продукта в базу

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 13 – Проверка введенных данных

Если проверка прошла успешно, то пользователь вводит цену и категорию продукта, которые проверяются на корректность данных при помощи цикла for и поэлементного сравнения символов (рисунок 14), например в стоимости не может быть букв (рисунок 15). Если же все проверки пройдены, то продукт добавляется в базу путем записи его в файл, иначе выдается сообщение об ошибке.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 14 – Проверка корректности цены и категории, а также вывод

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 15 – Неправильный ввод при добавлении

Пользователь также может удалить продукты из базы (рисунок 16). При удалении продукта из базы происходит предварительное считывание данных в список и вывод его элементов для более удобного использования. После чего пользователь вводит имя продукта, который он хочет удалить. Имя обрабатывается методом capitalize(), после чего проверяется на правильность ввода и наличие в базе. Если проверки пройдены, то совпадающее значение удаляется из списка при помощи метода pop() и остальной список записывается в файл, после чего выводится сообщение об успехе (рисунок 17). В противном случае пользователю выводится ошибка.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 16 – Функция удаления продукта из базы

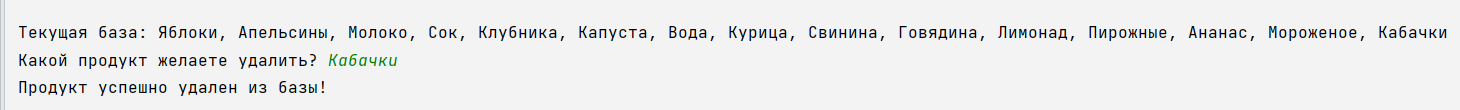


Рисунок 17 – Демонстрация удаления продукта

Функция просмотра выдает пользователю все продукты, записанные в базу (рисунок 18). Функция считывает данные из файла, после чего используется цикл for для построчного вывода продуктов в удобном для пользователя виде (рисунок 19).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 18 – Просмотр базы продуктовИзображение выглядит как текст

Автоматически созданное описание

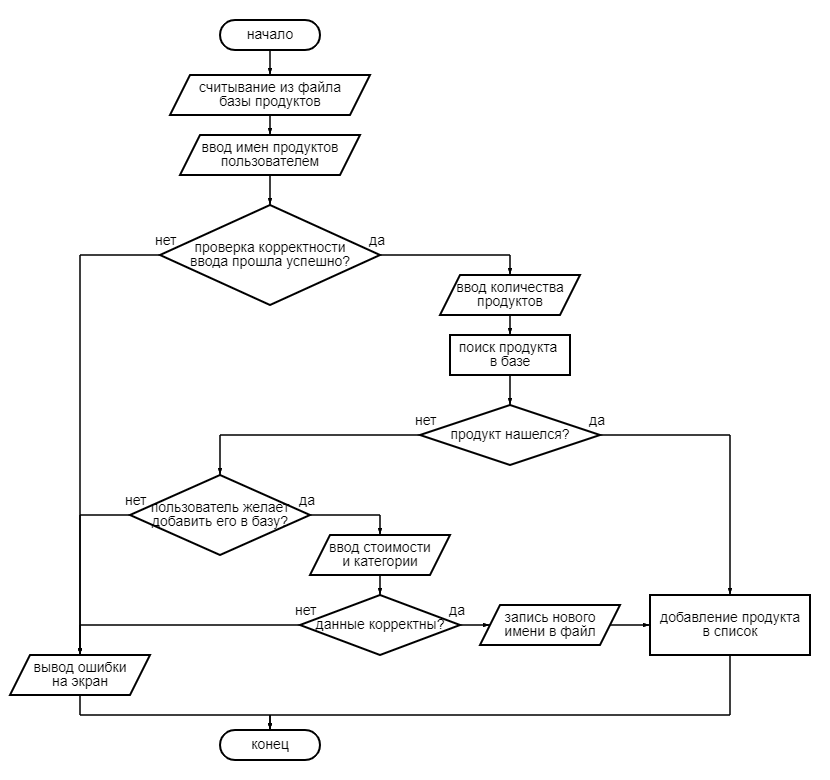
Рисунок 19 – Функция просмотра базы продуктов

1. Функция добавления продуктов в список, а также их удаления и просмотра. При вызове функции добавления в начале программа считывает данные из файла об уже записанных продуктах, далее пользователь вводит продукты, которые он хочет добавить в список, а также их количество. Если продукта нет в базе, то пользователю будет предложено его добавить туда, если пользователь согласится и все проверки будут пройдены, то продукт будет записан в базу и список (рисунок 20). Алгоритм добавления продуктов в список представлен на рисунке 21.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 20 – Демонстрация добавления продуктов в список

Рисунок 21 – Алгоритм добавления продуктов в список

Если пользователь хочет удалить что-то из списка, он может воспользоваться функцией удаления. Программа выведет все продукты, которые находятся в списке. Далее пользователь вводит продукт, который он хочет убрать, а также его количество, если он нашелся, иначе пользователю вернет ошибку. Если же продукт этого типа был один в список, то программа сразу удалить его из списка, не запрашивая количество у пользователя (рисунок 22).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 22 – Функция удаления одного продукта

Если же продуктов этого типа было несколько, то введенное пользователем число пройдет проверку на недопустимые символы при помощи посимвольного сравнения, а также проверку на нахождение в диапазоне (введенное число не может быть больше, чем количество продуктов в списке, а также меньше либо равно нулю). Далее программа записывает в отдельный список все продукты, которые надо удалить и после методом remove() убирает их из списка продуктов (рисунок 23), далее пользователю показывается конечный список (рисунок 24).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 23 – Функция удаления из списка нескольких продуктов

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 24 – Демонстрация удаления продуктов из списка

Пользователь также может просмотреть текущий список продуктов. Важно отметить, что функция будет выдавать ошибку, если список пуст. При выводе используется метод join(), который объединяет элементу в строку (рисунок 25), для более привычного восприятия (рисунок 26).



Рисунок 25 – Функция вывода списка



Рисунок 26 – Демонстрация вывода списка продуктов

1. Функция создания, просмотра и удаления записей о покупке. После того, как пользователь добавил продукты в список он может создать запись о покупке. При вызове функции создания записи программа считает данные о балансе из файла. Далее функция выведет сумму, которая будет необходима для «оплаты» и проверит достаточно ли у пользователя средств, если нет, то вернет ошибку (рисунок 27).

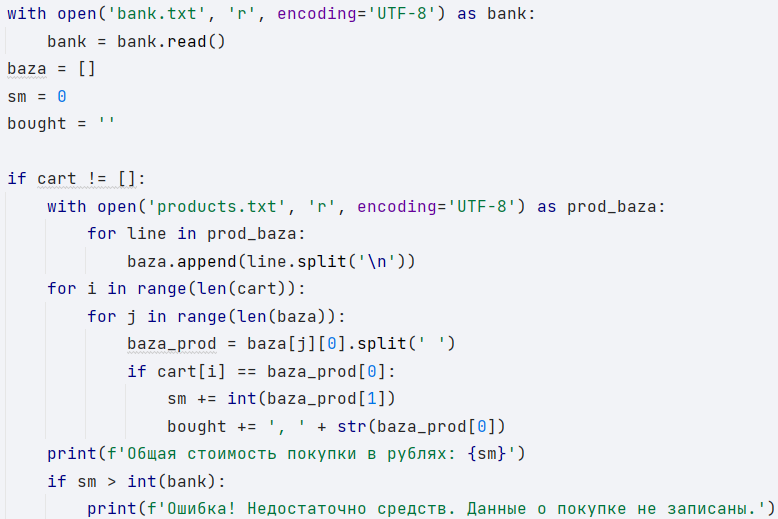


Рисунок 27 – Проверка средств для «оплаты»

Если средств хватает, то пользователю предоставляется выбор: ввести данные о дате вручную или позволить программе автоматически записать сегодняшнюю дату. В случае, если пользователь решил ввести дату вручную, то ему будет необходимо ввести год, месяц и день, которые записываются в отдельные переменные. Если все проверки пройдены, то при помощи библиотеки datetime будет записана эта дата. Важно сказать, что дата не будет записана, если она еще не наступила. Если же пользователь отказался вводить дату вручную, то с помощью datetime будет записана сегодняшняя дата. Если дата записана, то произойдет запись о покупке в файл, вычитание суммы из текущего баланса, а также очищение списка продуктов с помощью del (рисунок 28). После чего пользователю сообщится об успехе (рисунок 29).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 28 – Создание записи о покупке

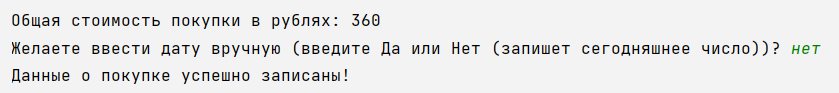


Рисунок 29 – Демонстрация создания записи о покупке

Пользователю также доступно удаление записей. При вызове этой функции программа выведет на экран считанные из файла пронумерованные записи, далее пользователю будет предложено ввести номер одной из записей (рисунок 30). Если введенный номер находится в диапазоне, то программа удалит эту запись из списка, а затем перезапишет файл. Более того, сумма, которая была потрачена на покупку, вернется и так же запишется в файл (рисунок 31).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 30 – Демонстрация удаления записи

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 31 – Функция удаления записи

Также пользователю доступен просмотр записей. Программа считывает записи из файла в список (рисунок 31), после чего выводит их в привычном пользователю виде (рисунок 32).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 31 – Функция просмотра записей

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 32 – Вывод записей о покупках

Пользователь может посмотреть записи по конкретной дате, категории или стоимости. Программа считывает записи в список, далее предлагает пользователю показать их по какому-то параметру. Если пользователь выбрал показ по дате, то ему необходимо ввести год, месяц и число. Далее если данные прошли проверку, то покажутся все записи за это число, если записей нет, то программа выведет соответствующее сообщение (рисунок 33).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 33 – Показ покупок по дате

Если пользователь выбрал показ по категории, то программа выведет все доступные категории при помощи считывания данных из файла в список, а далее применив set, чтобы избавиться от дубликатов. После чего пользователю будет доступен ввод категории и построчный вывод соответствующих записей, которые содержат эту категорию. В случае если записей нет, то пользователю выведется сообщение (рисунок 34).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 33 – Показ записей по категории

При показе по критерию стоимости пользователю предлагается ввести сумму, по которой будет произведен показ. После проверок программа выведет все записи с этой стоимостью или сообщение, что записей не нашлось (рисунок 34).

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 34 – Показ записей по стоимости

Для показа записей по дате и категории доступна сортировка по стоимости. Можно отсортировать по возрастанию или убыванию при помощи метода sort(), который берет для сравнения стоимость в записях (рисунок 35). Далее производится построчный вывод записей (рисунок 36). Полный алгоритм представлен на рисунке 37.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 35 – Сортировка по стоимости

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 36 – Демонстрация показа записей по дате с использованием сортировки

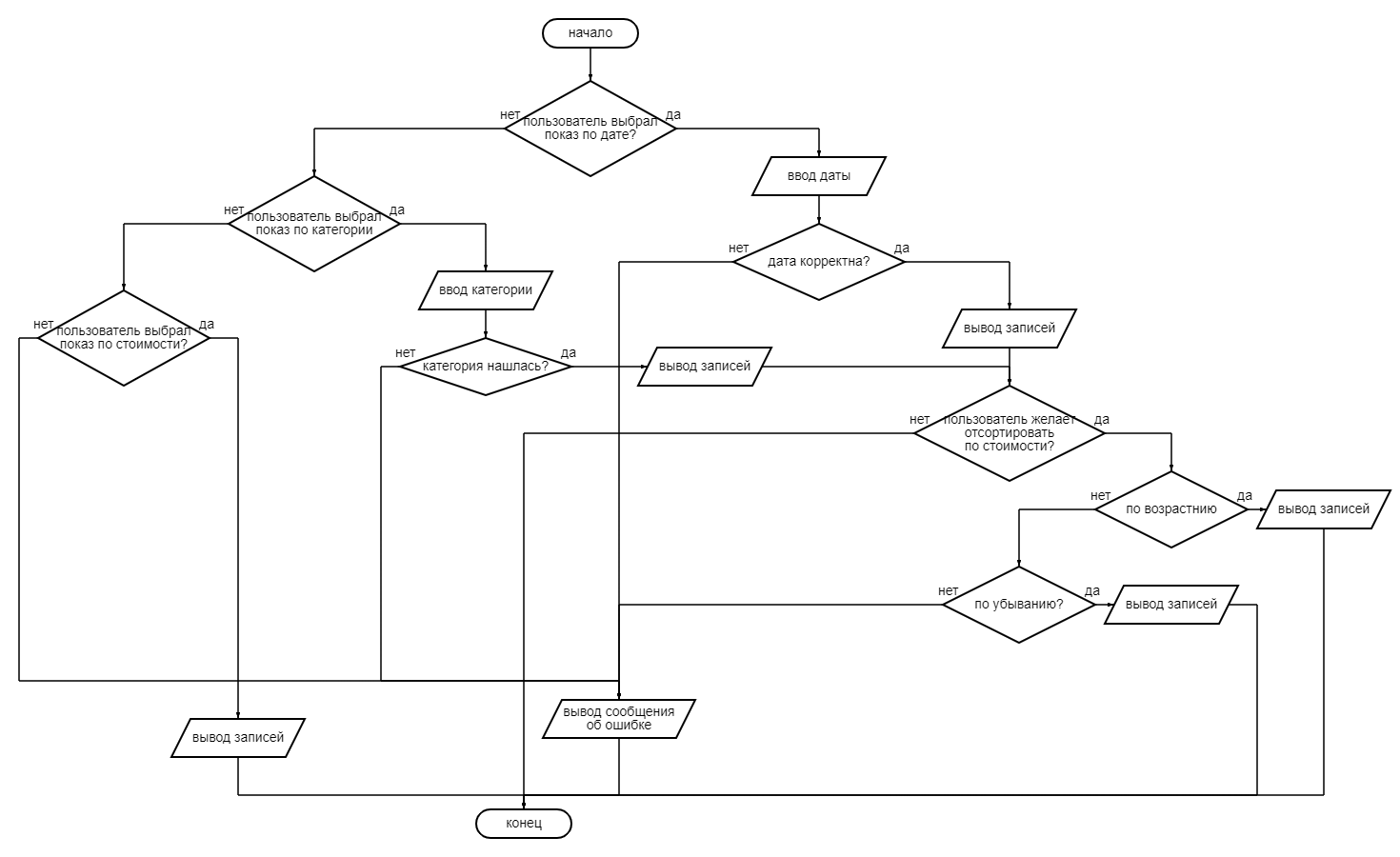


Рисунок 37 – Алгоритм показа записей по категориям

1. Функции интерфейса. В качестве интерфейса используется меню, где команды пронумерованы. Пользователь, вводя ту или иную цифру, перемещается по разным меню и выбирает действия (рисунок 38). Здесь так же предусмотрена проверка на корректность ввода, а также используется условный оператор (рисунок 39).

Изображение выглядит как стол

Автоматически созданное описание

Рисунок 38 – Демонстрация перемещения по различным меню

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 39 – Функция перемещения по меню

Когда пользователь закончил работу, он выбирает соответствующий пункт, после чего программа запускает соответствующую функцию (рисунок 40), прощается с пользователем и заканчивает свою работу (рисунок 41).

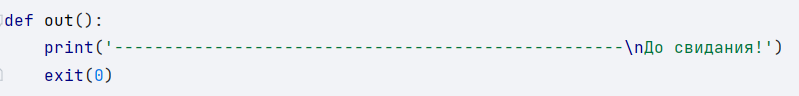


Рисунок 40 – Функция окончания работы программы

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 41 – Демонстрация окончания работы программы

**Вывод:**

В процессе выполнения домашнего задания был получен опыт создания программного обеспечения системы обработки данных на языке Python, что в дальнейшем поможет в создании более сложного программного обеспечения.