

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования «МИСиС – Национальный Исследовательский Технический Университет» НИТУ МИСиС

Институт Информационных Технологий и Компьютерных Наук
Кафедра АСУ

Ст. преподаватель :
Козлов М.Е.

«__» _____ 2021 г.

КУРСОВАЯ РАБОТА

Тема: «Разработка программного продукта»

Бакалавр: Белоусов Н.М.,

Группа: БИВТ-20-7

МОСКВА 2021

Содержание

3 - Введение

4 - Теоретическая часть

- Преимущества языка Java
- Недостатки языка Java
- Основные особенности языка Java

6 - Формат CSV

- Спецификация
- Стандартизация

8 - Описание задачи

8 - Описание используемых данных

9 - Разработка

10 - Декомпозиция кода

14 - Заключение

- Список литературы

Введение

С развитием информационных технологий и телекоммуникаций жизнь становится все более мобильной и информативной, новые технологии прочно входят в различные отрасли хозяйствования, сферы жизни и несут новые нормы в них. В связи с реформированием экономики, с взятием курса на инновационное развитие экономики, всё чаще и чаще в повседневной работе в большинстве предприятий и организаций используют различные средства информационно вычислительной техники и соответственно программного обеспечения. Но необходимо заметить, что спонтанное, не спланированное развитие в любой деятельности малоэффективно.

Исходя из этого цель данной курсовой работы - разработать программу, используя существующие знания для выполнения всех пунктов технического задания.

Теоретическая часть

Java — язык программирования общего назначения. Относится к объектно-ориентированным языкам программирования, к языкам с сильной типизацией. Java позволяет разработчикам приложений «написав однажды, запускать везде». Это означает, что скомпилированный Java-код может быть запущен на всех платформах, которые поддерживают Java, без необходимости в перекомпиляции.

Преимущества языка Java

- **Язык Java прост для изучения.** При разработке Java было уделено большое внимание простоте языка, поэтому программы на Java, по сравнению с программами на других языках, проще писать, компилировать, отлаживать и изучать.
- **Java - это объектно-ориентированный язык.** Это позволяет создавать модульные программы, исходный код которых может использоваться многократно.
- **Язык Java не зависит от платформы.** Одним из основных преимуществ языка Java является возможность переноса программ из одной системы в другую. Поскольку программы на Java не зависят от платформы как на уровне исходного кода, так и на двоичном уровне, их можно запускать в различных системах, что особенно важно для программ, предназначенных для World Wide Web.

Недостатки

- **Низкая производительность.** У любого языка высокого уровня довольно низкая производительность из-за компиляции и абстракции с помощью виртуальной машины.
- **Отсутствие нативного дизайна.** Для создания графического интерфейса пользователя (GUI) разработчики используют различные инструменты, ориентированные для конкретного языка.

Основные особенности языка Java

Простота

Надежность

Безопасность

Удобство

Чтобы запустить и использовать Java, вам нужно три вещи:

- JDK – Java Development Kit
- JRE – Java Runtime Environment
- JVM – Java Virtual Machine

Виртуальная машина Java (JVM) гарантирует, что у ваших приложений Java есть доступ к

минимальным ресурсам, необходимым для их запуска. Именно благодаря JVM программы

Java так легко запускаются на разных платформах.

Консольное приложение - это программа, которая не имеет графического интерфейса и которая работает в консоли. Команды в такой программе приходится вводить с клавиатуры, результаты работы консольные приложения также выводят на экран в текстовом режиме.

Формат CSV

CSV (от англ. Comma-Separated Values — значения, разделённые запятыми) — текстовый формат, предназначенный для представления табличных данных. Строка таблицы соответствует строке текста, которая содержит одно или несколько полей, разделённых запятыми.

Формат CSV стандартизирован не полностью. Идея использовать запятые для разделения полей очевидна, но при таком подходе возникают проблемы, если исходные табличные

данные содержат запятые или переводы строк. Возможным решением проблемы запятых и переносов строк является заключение данных в кавычки, однако исходные данные могут содержать кавычки. Помимо этого, термином «CSV» могут обозначаться похожие форматы, в которых разделителем является символ табуляции (TSV) или точка с запятой. Многие приложения, которые работают с форматом CSV, позволяют выбирать символ разделителя и символ кавычек.

Спецификация

- Каждая строка файла — это одна строка таблицы.
- Разделителем (англ. delimiter) значений колонок является символ запятой (,). Однако на практике часто используются другие разделители, то есть формат путают с DSV и TSV.
- Значения, содержащие зарезервированные символы (двойная кавычка, запятая, точка с запятой, новая строка) обрамляются двойными кавычками ("). Если в значении встречаются кавычки — они представляются в файле в виде двух кавычек подряд

Стандартизация

Название формата — CSV (значения, разделенные запятыми) — указывает что для разделения полей используется запятая. Тем не менее термин «CSV» может использоваться для обозначения схожих форматов. Много программ понимают под CSV более общий формат DSV (англ. delimiter-separated values — значения, разделённые разделителем), допускающий использование иных символов в качестве разделителя. В частности, в русской и других локалях запятая по умолчанию зарезервирована под десятичный разделитель. Поэтому как разделитель используется точка с запятой или табуляция (формат TSV). Другие отличия родственных форматов: используемые кавычки: двойные или одинарные; используемый набор символов; использование первой строки как заголовка. Это значительно затрудняет перенос данных из одних программ в другие, несмотря на всю простоту реализации поддержки CSV

Описание задачи:

Разработать консольное приложение со следующим функционалом:

- Работа с данными в csv файлах;
- Вывод результатов в csv файлы;
- Вывод ошибок в консоль;
- Регистронезависимый текстовый поиск по описанию транзакций;

Описание используемых данных

Таблица transactions.csv

Таблица содержит историю транзакций клиентов банка за один год и три месяца. Формат данных

```
customer_id,tr_datetime,mcc_code,tr_type,amount,term_id
111111,15 01:40:52,1111,1000,-5224,111111
111112,15 15:18:32,3333,2000,-100,11122233
...
```

Рисунок 1. Пример данных в файле transactions.csv

Описание полей:

- customer_id — идентификатор клиента;
- tr_datetime — день и время совершения транзакции (дни нумеруются с начала данных);
- mcc_code — МСС-код транзакции;
- tr_type — тип транзакции
- amount — сумма транзакции в условных единицах со знаком; + — начисление средств клиенту (приходная транзакция), - — списание средств (расходная транзакция);
- term_id — идентификатор терминала;

Разработка

Для более профессионального подхода к решению задачи было принято решение разделить программу на два файла: первый работает с файлом CSV, а второй - сама программа.

Язык программирования Java был выбран в силу того, что обладает простым синтаксисом, который позволяет писать гибкие и небольшие программы. Также на выбор языка повлиял большой набор как предустановленных, так и сторонних библиотек, предоставляющий набор инструментов для работы с базами данных, csv-файлами.

CSV файлы применяются для хранения огромных массивов данных, их очень удобно использовать при заполнении таблиц в базах данных.

Декомпозиция кода

Файл CsvFile содержит в себе следующие функции:

- CsvFile - читает его до конца
- getNames - возвращает имена в таблице
- search - ищет по индексу строку
- aveCsvFile - сохраняет файл
- toString - это функция возвращает строку

Файл Main содержит в себе следующие функции:

- Функция Main - сама логика программы
- Функция search - реализует сохранение
- readCsv - поиск и открытие файла
- readHeaders - выводит индекс и имя

В файле Main:

```
public class Main {  
    public static CsvFile csvFile;  
  
    public static void main(String[] args) {
```

Класс Main и метод main в нём запускает нашу программу

```

System.out.println("Добро пожаловать в программу по обработке csv файлов");
while (true) {
    System.out.println("\n1: Прочитать CSV файл");
    System.out.println("2: Просмотреть индексацию заголовков");
    System.out.println("3: Выполнить поиск");
    System.out.println("0: Выход");
    int choice = 0;
    Scanner in = new Scanner(System.in);
    try {
        choice = Integer.parseInt(in.next());
    } catch (NumberFormatException e) {
        System.out.println("В качестве команды могут быть только цифровые значения");
    }
}

```

Эта часть программы - цикл, который длится бесконечно(меню программы)

```

switch (choice) { case
    1:
        try {
            readCsv();
        } catch (IOException e) {
            System.out.println("К сожалению, произошла ошибка при чтении: " + e.getMessage());
        }
        break;
    case 2:
        if (csvFile != null) {
            readHeaders();
        } else {
            System.out.println("Сначала нужно прочитать файл");
        }
        break;
    case 3:
        if (csvFile == null) {
            System.out.println("Сначала нужно прочитать файл");
            break;
        }
        System.out.println("Введите имя файла, в который будут сохранены результаты");
        String fileName;
        String searchName;
        int index;
        fileName = in.next();
        System.out.println("Введите искомую подстроку");
        searchName = in.next();
        System.out.println("Введите индекс столбца (посмотреть можно в меню)");
        index = in.nextInt();
        try {
            search(fileName, searchName, index);
        } catch (IOException e) {
            System.out.println("Поиск не удался, причина: " + e.getMessage());
        }
        break;
    case 0:
        System.out.println("До свидания");
        return;
    default:
        System.out.println("Команда не существует");
}

```

Это логика программы, которая вызывает все нужные функции

```
public static void search(String fileName, String search, int index) throws IOException {  
    csvFile.saveCsvFile(csvFile.search(search, index), fileName);  
}
```

Эта функция сохраняет файл

```
public static void readCsv() throws IOException {  
    System.out.println("Введите файл и разделитель: ");  
    Scanner in = new Scanner(System.in);  
    csvFile = new CsvFile(in.next(), in.next());  
    System.out.println("Успешно прочитано");  
}
```

Эта функция считывает файл

```
public static void readHeaders() {  
    for (int i = 0; i < csvFile.getNames().length; i++) {  
        System.out.println(i + ": " + csvFile.getNames()[i]);  
    }  
}
```

Эта функция выводит результат

В файле CsvFile:

```
CsvFile(String fileName, String separator) throws IOException {
    this.separator = separator;
    BufferedReader reader = new BufferedReader(new FileReader(fileName));
    String line = null;
    line = reader.readLine();
    names = line.split(separator);
    while ((line = reader.readLine()) != null) {
        lines.add(line.split(separator));
    }
}
```

Данный метод читает файл до конца заменяя пробелами разделяющий символ

```
public CsvFile search(String str, int index) {
    if (index >= names.length || index < 0) {
        throw new IllegalArgumentException("wrong csv index: " + index);
    }
    CsvFile csvFile = new CsvFile();
    csvFile.separator = this.separator;
    csvFile.names = names;
    for (int i = 0; i < lines.size(); i++) {
        if (lines.get(i)[index].toLowerCase().contains(str.toLowerCase())) {
            csvFile.lines.add(lines.get(i));
        }
    }
    return csvFile;
}
```

Данная функция ищет по индексу нужную нам строку.

```
public void saveCsvFile(CsvFile csvFile, String filename) throws IOException {
    FileWriter fileWriter = new FileWriter(filename);
    for (int i = 0; i < csvFile.names.length; i++) {
        fileWriter.write(csvFile.names[i]);
        if (i != (csvFile.names.length - 1)) {
            fileWriter.write(csvFile.separator);
        }
    }
    fileWriter.write(System.getProperty("line.separator"));
    for (String[] line : csvFile.lines) {
        for (int i = 0; i < line.length; i++) {
            fileWriter.write(line[i]);
            if (i != (line.length - 1)) {
                fileWriter.write(csvFile.separator);
            }
        }
        fileWriter.write(System.getProperty("line.separator"));
    }
}
```

Данная функция сохраняет файл и записывает в него нужные данные

Заключение

В рамках выполнения данной курсовой работы было разработано консольное приложение с использованием базы данных.. В ходе выполнения курсовой работы была произведена работа с данными в csv-файлах, базами данных и языком Java.

В приложении предусмотрен отлов большинства прогнозируемых ошибок, которые могут возникнуть в результате работы программы, что обеспечивает безопасное использование программы, а также сохранение данных в безопасности.

Список литературы

- 1) Помощь при нахождении багов/ошибок <https://stackoverflow.com>
- 2) SQL + Java <https://javarush.ru/groups/posts/1952-vvedenie-v-sql>
- 3) Руководство по языку программирования Java <https://metanit.com/java/tutorial/>
- 4) С.А. Орлов «Программная инженерия. Технологии разработки программного обеспечения». 5ое издание обновленное и дополненное – СПб.: Питер, 2016 – 640 с.
- 5) Г. Шилдт «Java. Полное руководство» 10-е издание - Диалектика-Вильямс, 2018 – 1488 с.
- 6) Ю. Козмина, Р. Харроп «Spring 5 для профессионалов» - Диалектика-Вильямс, 2019 - 1120 с.
- 7) К. Уоллс. «Spring в действии», 2014 – 624 с.