

Universidade do Minho
Departamento de Informática

Mestrado Integrado em Engenharia Informática

Inteligência Ambiente: Tecnologias e Aplicações



Questão-Aula Nº1 - Rule-based Automotive Control System

a86617 Gonçalo Nogueira

Braga
Dezembro, 2020

Conteúdo

1	Introdução	3
2	Objetivos	4
3	Dados recolhidos e tratamento	5
3.1	Leitura do <i>dataset</i>	5
3.2	Tratamento dos dados	7
3.3	Escrita dos resultados num <i>log</i>	9
4	Conclusão	11

Lista de Figuras

1	Abertura de ficheiro com nome <i>"Anexo-temperatura.xlsx"</i>	5
2	Conversão em objeto <i>XLSXWorkbook</i> e obtenção da primeira folha	5
3	Iteração por cada linha coluna a coluna	6
4	Método <i>checkSeason</i> criado	7
5	Classe <i>Handler</i> criada	7
6	Método <i>daDifTemp</i>	8
7	Criação do <i>file</i> e do <i>printWriter</i>	9
8	Método <i>toString()</i>	9
9	Exemplo de uma entrada no log	10

1 Introdução

No âmbito da unidade curricular de Inteligência Ambiente: Tecnologias e Aplicações foi proposto aos alunos a construção de um sistema baseado em regras (*SBR*) no contexto de ambientes inteligentes e análise e tratamento de dados recolhidos de sensores.

O sistema proposto aos alunos seria um *ACS*, acrónimo de *Rule-based Automotive Control System*, em específico o tratamento de dados sensoriais de temperatura para controlo e ajuste automático do sistema de ar-condicionado de um veículo.

2 Objetivos

- Implementar um sistema capaz de monitorizar leituras de sensores físicos.
- O sistema deverá ser capaz de analisar e tratar os dados recolhidos do sensor utilizado.
- Depois do tratamento deverá escrever num *logger* o ajuste de temperatura do ar-condicionado.

3 Dados recolhidos e tratamento

3.1 Leitura do *dataset*

No ano anterior para a unidade curricular de *Sistemas de Representação de Conhecimento e Raciocínio* aquando da resolução de um trabalho prático criei um *Parser* para a leitura de um ficheiro *.xlsx*. Com vista a reutilizar parte do programa desenvolvido e devido á minha familiarização com esse assunto resolvi converter o ficheiro fornecido de *.csv* para *.xlsx*.

Tal como no projeto desenvolvido na altura, utilizei a biblioteca *POI* na linguagem *Java*. Esta biblioteca permite a leitura e manipulação de ficheiros *Excell* de uma forma muito rápida e conveniente.

Na parte do código em si, para abrir o ficheiro basta abrir o ficheiro no programa (Figura 1) e depois transformar esse ficheiro num objeto *XSSFWorkbook* (Figura 2) que é similar nas suas propriedades a um ficheiro *Excell* e por isso temos de ir buscar a primeira *Sheet* utilizando o método *getSheetAt(0)* que devolve um objeto do tipo *XSSFSheet* também já definido na biblioteca (Figura 2).

Em seguida, percorro cada linha coluna a coluna utilizado também métodos presentes na biblioteca (Figura 3).

```
File excelFile = new File( pathname: "Anexo-temperatura.xlsx");
FileInputStream fis = new FileInputStream(excelFile);
```

Figura 1: Abertura de ficheiro com nome "Anexo-temperatura.xlsx"

```
// we create an XSSFWorkbook object for our XLSX Excel File
XSSFWorkbook workbook = new XSSFWorkbook(fis);
// we get first sheet
XSSFSheet sheet = workbook.getSheetAt( index: 0);
```

Figura 2: Conversão em objeto *XLSXWorkbook* e obtenção da primeira folha

```

for (Row row : sheet) {
    // iterate on cells for the current row
    Iterator<Cell> cellIterator = row.cellIterator();
    pw.println("Dados= { ");
    while (cellIterator.hasNext() && flag<6 ) {
        Cell cell = cellIterator.next();
        if(flag==0) a.setEstacao(checkSeason(LocalDate.parse(cell.toString()),formatter));
        if(flag==4) a.setTemp(Float.parseFloat(cell.toString()));
        pw.println(cell.toString()+",");
        System.out.print(cell.toString() + ",");
        flag++;
    }
}

```

Figura 3: Iteração por cada linha coluna a coluna

3.2 Tratamento dos dados

Relativamente aos dados, á medida que faço as iterações coluna a coluna vou convertendo a data usando a função *checkSeason* (Figura 4) que devolve **1** caso seja outono ou inverno e **2** caso seja primavera ou verão e a temperatura lida em cada linha para um objeto Handler criado para tratamento desses dados (Figura 5).

Dentro do *handler* criei o método *daDifTemp* que é responsável por consoante a estação presente na instância da classe devolver em *String* o valor a colocar no ar-condicionado, comparando a temperatura lida á temperatura de conforto, verificando também se é inferior ou superior (Figura 6).

```
private static int checkSeason(LocalDate data) {  
    int a=0;  
    if ((data.getDayOfMonth() > 22 && data.getMonthValue() > 9) || (data.getDayOfMonth() < 20 && data.getMonthValue() < 3))  
        a = 1; // outono+inverno  
    if ((data.getDayOfMonth() >= 20 && data.getMonthValue() >= 3) || (data.getDayOfMonth() <= 22 && data.getMonthValue() <= 9))  
        a = 2; // verao+primavera  
    return a;  
}
```

Figura 4: Método *checkSeason* criado

```
public class Handler {  
    float temp;  
    int estacao;
```

Figura 5: Classe *Handler* criada


```

public String daDifTemp(float temp){
    float airCond=0;
    float conforto=0;
    String s=null;
    if(estacao==1){
        conforto=15;
        if(temp>=conforto){
            airCond=temp-conforto;
            s="airconditioning-{" + airCond + "}";
        } else {
            airCond=conforto-temp;
            s="airconditioning+{" + airCond + "}";
        }
    }
    if(estacao==2){
        conforto=25;
        if(temp>=conforto){
            airCond=temp-conforto;
            s="airconditioning-{" + airCond + "}";
        } else {
            airCond=conforto-temp;
            s="airconditioning+{" + airCond + "}";
        }
    }
    return s;
}

```

Figura 6: Método *daDifTemp*

3.3 Escrita dos resultados num *log*

Depois da análise e tratamento de dados é necessário fazer o output dos resultados para um *log*. Para o efeito criei um ficheiro texto e utilizei um *printWriter* para escrever nesse ficheiro, sendo este método bastante cómodo em *Java* pois é feito da mesma forma que se faz output para a tela só que neste caso vai para o ficheiro criado (Figura 7).

Ao mesmo tempo que vou percorrendo a linha pelas suas colunas imprimo essa informação no log para ficar registado os valores lidos. No fim da linha imprimo para o ficheiro a instância do objeto *Handler* criado chamando o método *toString()* (Figura 8) definido na classe que apresenta os resultados após o tratamento de dados.

Depois de correr o programa desenvolvido o ficheiro log contém o formato apresentado na figura 9.

```
FileWriter fw = new FileWriter( fileName: "logger.pl");  
PrintWriter pw = new PrintWriter(fw);
```

Figura 7: Criação do *file* e do *printWriter*

```
@Override  
public String toString() {  
    return "-> Estação= " + daEstacao(estacao) + " " + daDifTemp(temp) ;  
}
```

Figura 8: Método *toString()*

```
Dados= {  
  2012-01-01 00:00:00,  
  Guimaraes,  
  41.44253,  
  -8.291786,  
  9.88,  
  91.0,  
}  
-> Estação= verao/primavera aircondtioneing+{15.12}
```

Figura 9: Exemplo de uma entrada no log

4 Conclusão

Tendo em conta os objetivos que me propus a realizar, penso que posso concluir que os cumprí na íntegra pois consegui apresentar no ficheiro log os valores da forma que me foram pedidos.

Não encontrei nenhuma grande dificuldade durante toda a conceção do sistema visto a simplicidade do projeto proposto.