DIFCUL
LabP 2013/2014
Trabalho 4 – A impressora
Ana Respício e Jorge Gomes

O sistema de *spooling*, que controla a impressão de ficheiros de uma impressora, pode estar concretizado considerando diferentes prioridades e usando diferentes políticas de impressão. Neste projecto, pretendemos ver a ordem de impressão de acordo com três políticas diferentes. Para esse efeito, considere um sistema de impressão de ficheiros com as características que se descrevem de seguida.

Cada ficheiro tem os atributos:
- Nome do ficheiro;
- Tamanho (em KB);
- Utilizador (nome do utilizador que enviou o ficheiro).

O sistema de *spooling* desta impressora considera 3 níveis de prioridade de impressão, onde:
- os ficheiros enviados pelo utilizador "admin" têm sempre a prioridade mais elevada (2),
- para os ficheiros dos restantes utilizadores, a prioridade é definida pela pelo tamanho de acordo com a seguinte tabela. Por exemplo, um ficheiro com 87KB tem prioridade 1 e um com 43KB tem prioridade 2.

Prioridade	Tamanho mínimo do ficheiro (KB)	Tamanho máximo do ficheiro (KB)
2	0	64
1	65	1024
0	1025	Não definido

A impressora tem uma determinada capacidade de memória para armazenar ficheiros em espera (dada como argumento). Ao inserir um novo ficheiro em espera para impressão deve ser verificado se há memória suficiente. Se o tamanho do ficheiro exceder a capacidade livre da impressora, o ficheiro não é colocado em espera, gera-se uma mensagem de erro correspondente, e a impressora continua a receber outros pedidos de impressão.

Políticas de impressão

As políticas de impressão definem a ordem pela qual os ficheiros são removidos da impressora (até ficar vazia). Em caso de empates na ordem de impressão, deve ser sempre impresso primeiro o mais antigo (isto é, que foi metido primeiro).

Política 1:
1. todos ficheiros com prioridade 2;
2. todos ficheiro com prioridade 1;
3. todos ficheiros com prioridade 0.

Política 2:

1. Repete o seguinte até não haver ficheiros de prioridade 1 ou 2:
    a. dois ficheiros com prioridade 2;
    b. um ficheiro com prioridade 1;
2. os ficheiros com prioridade 0 apenas são impressos quando não existem ficheiros pendentes com prioridades superiores.

Política 3:

- ignora todas as prioridades e imprime pela ordem que os ficheiros foram submetidos.

Objectivos do projecto

Pretende-se desenvolver uma aplicação que, dada uma impressora (definida por uma política de impressão e uma capacidade) e uma sequência de ficheiros enviados para a impressora, nos permita ver a ordem da sua impressão. Esta aplicação deve fazer uso dos tipos de dados *Queue* e *Printer*, a desenvolver pelos alunos.

Queue

Considere uma implementação de uma *Queue* baseada numa lista simplesmente ligada. Deve recorrer às classes apresentadas em AED, que se disponibilizam junto com este enunciado,
https://www.di.fc.ul.pt/~mal/aed/queue/src/Queue.java e
https://www.di.fc.ul.pt/~mal/aed/queue/src/SLLQueue.java, e completá-las.

Printer

Este tipo deve fazer uso do tipo *Queue*, concretamente para implementar uma (ou mais) **fila de ficheiros**. Deve considerar o tipo *PrinterTask* que representa um ficheiro para impressão e cujo *.java* está anexo a este enunciado.

A classe *Printer* deve disponibilizar as seguintes operações:

- public Printer(int policy, int capacity) – construtor – cria uma Printer com a política e capacidade dadas.
- public boolean enqueue(PrinterTask task) – insere um ficheiro na impressora, na fila da prioridade correspondente; retorna false se task não puder ser colocado em espera.
- public boolean isEmpty() – retorna true se e só se esta impressora está sem ficheiros em espera.
- public PrinterTask takeNext() – retorna a próxima tarefa de impressão (ficheiro) a ser processada, de acordo com a política da impressora, removendo-o de espera. Deve tomar atenção a pré-condições.
- public int availableSpace() – retorna o espaço de memória disponível (em KB).
- public String toString() – retorna a informação do conteúdo da impressora numa String com o seguinte formato
  politica: capacidade: capDisponivel
  para cada fila q: q.toString()

O que devem fazer

Implementar os tipos descritos acima.

Importante: As assinaturas dos métodos públicos das classes Printer, Queue, e SLLQueue têm que ser respeitadas. A classe PrinterTask não deve ser modificada.

Os tipos devem ser testados com uma aplicação cliente RunPrinter passando como argumentos na linha de comando:

- o nome de um ficheiro que contém a informação dos ficheiros a imprimir;
- a política da impressora;
- a capacidade da impressora em KB.

Por exemplo, o comando:

```
> java RunPrinter dados.txt 1 16000
```

deve executar uma impressora com a política 1 e 16.000KB de capacidade e a sequência de ficheiros em dados.txt.

Cada linha do **ficheiro de dados** contém a informação de uma tarefa de impressão:

```
nome tamanho utilizador
```

A classe RunPrinter deve testar a classe Printer usando os seguintes passos:

1. Criar uma Printer com a política (args[1]) e capacidade (args[2]) dadas,
2. Ler o ficheiro de dados (args[0]), e por cada linha lida, fazer enqueue da tarefa de impressão (ficheiro) aí descrita; Caso uma dada tarefa (ficheiro) não possa ser inserida na impressora, escrever uma mensagem de erro para o ecrã.
3. Pôr a impressora a funcionar: remover todas as tarefas pendentes de acordo com a política de impressão (1, 2 ou 3) e por cada tarefa de impressão, que remove, escreve no ecrã uma linha com a sua informação no formato
   nome: tamanho: utilizador.

Entrega

Desenvolva as classes `SLLQueue.java`, `Printer.java` e `RunPrinter.java`. Deve criar o ficheiro trab4.zip, contendo os ficheiros .java, com a instrução

```
zip trab4.zip PrinterTask.java SLLQueue.java Printer.java RunPrinter.java
```

Este ficheiro deverá ser submetido através do formulário de submissão.

Exemplos de input/output

dados.txt 1 16000
 	
`usaLista.java 2 estroncio`
`listaAlunos.txt 34 admin`
`notas.xls 1024 falancio`
`inverte 1 czar`
`historico.xls 3000 admin`
`lista.java 3 erroncio`
`lista.h 2 erroncio`
`usaLista.java 3 erroncio`
`historia 5 hermano`
`relatorio.doc 2048 pancracio`
`mail 1000 estroncio`
`copia.doc 1205 cabulacio`
`arvore.h 10 cabulacio`
`arvore.java 25 cabulacio`
	Resultado para dados.txt 1 16000
	`Sequencia de impressao:`
	`usaLista.java:2:estroncio`
	`listaAlunos.txt:34:admin`
	`inverte:1:czar`
	`historico.xls:3000:admin`
	`lista.java:3:erroncio`
	`lista.h:2:erroncio`
	`usaLista.java:3:erroncio`
	`historia:5:hermano`
	`arvore.h:10:cabulacio`
	`arvore.java:25:cabulacio`
	`notas.xls:1024:falancio`
	`mail:1000:estroncio`
	`relatorio.doc:2048:pancracio`
	`copia.doc:1205:cabulacio`

dados.txt 1 4000 (como o anterior mas com capacidade 4000)
 	
`usaLista.java 2 estroncio`
`listaAlunos.txt 34 admin`
`notas.xls 1024 falancio`
`inverte 1 czar`
`historico.xls 3000 admin`
`lista.java 3 erroncio`
`lista.h 2 erroncio`
`usaLista.java 3 erroncio`
`historia 5 hermano`
`relatorio.doc 2048 pancracio`
`mail 1000 estroncio`
`copia.doc 1205 cabulacio`
`arvore.h 10 cabulacio`
`arvore.java 25 cabulacio`
	Resultado para dados.txt 1 4000
	`O ficheiro historico.xls nao foi colocado em espera. Espaco disponivel: 2939`
	`O ficheiro mail nao foi colocado em espera. Espaco disponivel: 878`
	`O ficheiro copia.doc nao foi colocado em espera. Espaco disponivel: 878`
	`Sequencia de impressao:`
	`usaLista.java:2:estroncio`
	`listaAlunos.txt:34:admin`
	`inverte:1:czar`
	`lista.java:3:erroncio`
	`lista.h:2:erroncio`
	`usaLista.java:3:erroncio`
	`historia:5:hermano`
	`arvore.h:10:cabulacio`
	`arvore.java:25:cabulacio`
	`notas.xls:1024:falancio`
	`relatorio.doc:2048:pancracio`

dados.txt 2 16000 (agora com a política 2)
 	
`usaLista.java 2 estroncio`
`listaAlunos.txt 34 admin`
`notas.xls 1024 falancio`
`inverte 1 czar`
`historico.xls 3000 admin`
`lista.java 3 erroncio`
`lista.h 2 erroncio`
`usaLista.java 3 erroncio`
`historia 5 hermano`
`relatorio.doc 2048 pancracio`
`mail 1000 estroncio`
`copia.doc 1205 cabulacio`
`arvore.h 10 cabulacio`
`arvore.java 25 cabulacio`
	Resultado para dados.txt 2 16000
	`Sequencia de impressao:`
	`usaLista.java:2:estroncio`
	`listaAlunos.txt:34:admin`
	`notas.xls:1024:falancio`
	`inverte:1:czar`
	`historico.xls:3000:admin`
	`mail:1000:estroncio`
	`lista.java:3:erroncio`
	`lista.h:2:erroncio`
	`usaLista.java:3:erroncio`
	`historia:5:hermano`
	`arvore.h:10:cabulacio`
	`arvore.java:25:cabulacio`
	`relatorio.doc:2048:pancracio`
	`copia.doc:1205:cabulacio`

dados.txt 3 16000 (agora com a política 3)
 	
`usaLista.java 2 estroncio`
`listaAlunos.txt 34 admin`
`notas.xls 1024 falancio`
`inverte 1 czar`
`historico.xls 3000 admin`
`lista.java 3 erroncio`
`lista.h 2 erroncio`
`usaLista.java 3 erroncio`
`historia 5 hermano`
`relatorio.doc 2048 pancracio`
`mail 1000 estroncio`
`copia.doc 1205 cabulacio`
`arvore.h 10 cabulacio`
`arvore.java 25 cabulacio`
	Resultado para dados.txt 3 16000
	`Sequencia de impressao:`
	`usaLista.java:2:estroncio`
	`listaAlunos.txt:34:admin`
	`notas.xls:1024:falancio`
	`inverte:1:czar`
	`historico.xls:3000:admin`
	`lista.java:3:erroncio`
	`lista.h:2:erroncio`
	`usaLista.java:3:erroncio`
	`historia:5:hermano`
	`relatorio.doc:2048:pancracio`
	`mail:1000:estroncio`
	`copia.doc:1205:cabulacio`
	`arvore.h:10:cabulacio`
	`arvore.java:25:cabulacio`