

Programação 1 (LTI), 2018/2019

Projeto

(este enunciado tem 15 páginas)

iCageDoree

0. Contexto

Realizado em 2013 pelo luso-descendente Ruben Alves, o filme *La Cage Dorée* (A Gaiola Dourada) tornou-se um sucesso de bilheteira ao apresentar sob a forma de comédia a vida de um casal típico de imigrantes portugueses, a Maria e o José Ribeiro, a viver em Paris, em que a Maria é *concierger* (porteira) e o José *maçon* (pedreiro). O filme gira em torno das atribulações deste casal que é chamado a resolver os mais variados problemas da vida quotidiana dos seus vizinhos, desde levar crianças à escola até reparar fugas de água na canalização.



Inspirado por este filme, cinco anos mais tarde, um outro luso-descendente, François Martins, profissional das tecnologias da informação, encontra-se agora na fase final do lançamento da sua *startup* em Paris, a que deu o nome sugestivo de iCageDoree. O modelo de negócio desta empresa baseia-se em colocar em contacto profissionais especialistas em arranjos domésticos (e.g. reparação de fogões, instalação de ar condicionado, substituição de fechaduras, renovação de estores, etc) e clientes destes especialistas através de uma plataforma online.

Para tanto precisará de recorrer a um software a desenvolver, batizado de cyberConcierge, que assegure a gestão dos pedidos de contacto pelos clientes, dos especialistas e do emparelhamento entre uns e outros.

Por exemplo, um pedido de ajuda sobre ferros de engomar deve ser direcionado para um profissional especializado nesta área, e não na área de televisores; um pedido por um cliente de Paris deve ser direcionado para um especialista dessa cidade e não de Lisboa, etc.

Essa é a aplicação cujo núcleo vai ser desenvolvido no presente exercício pedagógico de programação.

1. Software a desenvolver

Objetivo

Com uma finalidade pedagógica, usando Python 3, neste projeto vai desenvolver o software *cyberConcierge*, usado pela empresa *iCageDoree* para gerir a atribuição de pedidos de ajuda, feitos por clientes, aos especialistas seus colaboradores.

Funcionalidade

O seu programa **recebe** uma listagem dos especialistas que caracteriza, num dado momento, cada um dos especialistas quanto a aspetos relevantes para o seu atendimento de pedidos de ajuda/contratação feitos por clientes. Recebe também uma listagem dos pedidos de ajuda/contratação que se encontram por atribuir a especialistas até esse mesmo momento.

O seu programa **entrega**, por um lado, o um plano de atendimento pelos especialistas desses pedidos de ajuda dos clientes. Por outro lado, entrega ainda a listagem atualizada dos especialistas, após os pedidos de ajuda terem sido distribuídos por eles.

Entrada

O programa recebe ficheiros com nomes e estruturas internas para arrumação de informação similares à dos seguintes exemplos fragmentários:

2018y10m30experts10h30.txt

```
Day:
2018-10-30
Time:
10h30
Company:
iCageDoree
Experts:
Dan Tufis, lisbon, (heating; doors; windows), 2*, 20, 2018-10-30, 09:00,
2879.0
Nelo Martins, lisbon, (lockers), 5*, 100, 2018-10-30, 18:00, 22045.0
Paulo Borges, paris, (electricity; doors), 3*, 65, 2018-10-31, 12:30, 2204.0
Lech Walesa, london, (plumbing), 5*, 50, 2018-11-01, 10:00, 3523.0
...
Marko Tadjic, ...
```

Os especialistas estão ordenados, de cima para baixo, desde o que estará há mais tempo disponível para o que estará há menos tempo. Cada especialista é caracterizado numa única linha.

2018y10m30clients10h30.txt

```
Day:
2018-10-30
Time:
10:30
Company:
iCageDoree
Clients:
Guillaume Dutroux, paris, 2018-11-01, 13:30, 45, 4*, plumbing, 4h00
Jose Quesada, madrid, 2018-10-31, 10:00, 30, 2*, refrigerator, 5h15
Martin Wyne, london, 2018-10-30, 19:00, 105, 5*, wifi, 0h30
...
```

Os clientes estão ordenados, de cima para baixo, desde o que fez o pedido há mais tempo para o que fez há menos tempo.

Saída

O programa produz dois ficheiros, um com a listagem dos especialistas atualizada e outro com a calendarização da execução das obras. Com uma finalidade pedagógica, assume-se que a atualização e a calendarização são feitas em simultâneo e de trinta em trinta minutos.

O ficheiro de saída com a listagem de especialistas atualizada tem uma estrutura interna similar ao ficheiro de entrada com a listagem dos especialistas, em que cada especialista é caracterizado numa única linha. A diferença é que o cabeçalho é atualizado quanto à data e ao tempo (incrementado de 30 minutos em relação ao momento do ficheiro de entrada), e os três últimos campos de cada especialista — dois respetivamente com a data e a hora em que ficará liberto para atender novo cliente não calendarizado, e um último campo com o montante (em euros) acumulado — são atualizados em função da calendarização feita para os pedidos de ajuda/contratação.

O ficheiro com a calendarização da execução dos pedidos de contratação tem uma estrutura interna para arrumação de informação similar à do seguinte exemplo fragmentário:

2018y10m30schedule11h00.txt

```
Day:
2018-10-30
Time:
11:00
Company:
iCageDoree
Schedule:
2018-10-30, 16:00, Maria Schwartz, Maciej Salawa
2018-10-30, 16:15, John Smith, Carlos Silva
...
```

Mais sobre especificação

- As diferentes listagens (especialistas, clientes, calendarização) são guardadas em ficheiros .txt.
- Cada listagem começa com um cabeçalho que contém a indicação do dia de operação, da hora de operação, da empresa e do âmbito do ficheiro (*viz.* Experts, Clients ou Schedule) como neste exemplo para o caso de uma listagem de clientes:

Day:
2018-10-30
Time:
10:30
Company:
iCageDoree
Clients:

- Cada ficheiro de entrada e de saída é nomeado de acordo com a seguinte convenção: concatenação das strings que designam o ano, "y", o mês, "m", o dia, o âmbito do ficheiro, a hora, "h", os minutos, ".txt", em minúsculas, como neste exemplo referente ao ficheiro com o cabeçalho do ponto anterior:

2018y10m30clients10h30.txt

- Na **listagem de especialistas**, a seguir ao cabeçalho, cada linha corresponde a um especialista (cujos respetivos elementos informativos estão separados por vírgulas) estando a listagem ordenada de cima para baixo por ordem da sucessiva disponibilidade do especialista (e depois por ordem alfabética dos nomes dos operadores em caso de empate). Cada especialista é caracterizado por uma única linha com:
 - nome (e.g. Dan Tufis),
 - zona/cidade de trabalho (e.g. lisbon),
 - domínios da sua especialidade, entre parêntesis e separados por ponto e vírgula (e.g. (heating; doors; windows)),
 - reputação representada em número de estrelas (e.g. 2*) de 1* a 5*,
 - preço da hora de trabalho, em euros (e.g. 20),
 - data de finalização do último pedido que vai atender (e.g. 2018-10-30),
 - hora de disponibilidade, que é a hora da finalização do último pedido que vai atender (e.g. 9:00) (n.b.: para atribuição de nova tarefa ao especialista, contará a mais tardia de entre esta data e a data do cabeçalho),
 - e montante acumulado recebido pelo trabalho já realizado (incluindo todos os que estão calendarizados),

como no seguinte exemplo:

Dan Tufis, lisbon, (heating; doors; windows), 2*, 20, 2018-10-30, 9:00, 2879.0

O cabeçalho indica a data e a hora da última atualização da listagem dos especialistas. Com uma finalidade pedagógica, assume-se que as atualizações são feitas de trinta em trinta minutos.

- Na **listagem de clientes em espera**, a seguir ao cabeçalho, cada linha corresponde a um pedido de ajuda/contratação do trabalho de um perito (cujos elementos informativos estão separados por vírgulas). Os pedidos estão ordenados, de cima para baixo, pela ordem de chegada (atenção: não pela ordem das datas/horas para que a sua execução é pedida). Cada pedido é caracterizado por uma única linha com:
 - o cliente que fez o pedido (e.g. Jose Quesada),
 - a zona/cidade para onde o pedido é feito (e.g. madrid),
 - a data a partir da qual o pedido deve ser executado (e.g. 2018-10-31) e a hora (e.g. 10:00),
 - o máximo que o cliente está disposto a pagar por hora de trabalho (e.g. 30),
 - o mínimo de reputação do especialista pretendido (e.g. 2*),
 - o domínio para que é pedido o trabalho do especialista (e.g. refrigerator),
 - o tempo durante o qual o cliente quer contratar o especialista (e.g. 5h15),

como no seguinte exemplo:

Jose Quesada, madrid, 2018-10-31, 10:00, 30, 2*, refrigerator, 5h15

Para efeitos pedagógicos vamos considerar a duração do trabalho requisitado fixada de antemão pelo cliente.

O cabeçalho indica a data e a hora desde que têm estado a ser registados os pedidos, que devem ser idênticas à data e hora da última atualização do ficheiro de entrada dos especialistas.

- Na **calendarização do atendimento dos pedidos de contratação**, a seguir ao cabeçalho, cada linha corresponde ao atendimento de um pedido calendarizado (cujos elementos informativos estão separados por vírgulas) estando a listagem ordenada por ordem crescente do momento de atendimento. Cada atendimento calendarizado é caracterizado pela data (e.g. 2018-10-30) e hora de início (e.g. 16:00), pelo nome do cliente que fez o pedido (e.g. Maria Schwartz), e pelo nome do especialista que vai responder a esse pedido (e.g. Maciej Salawa), como ilustrado no seguinte exemplo:

2018-10-30, 16:00, Maria Schwartz, Maciej Salawa

A hora de início é a hora mais tardia entre a hora de início pedida pelo cliente e a hora de disponibilidade do especialista.

O cabeçalho é similar aos dos ficheiros de entrada, atualizado quanto ao tempo (incrementado de trinta minutos em relação ao momento dos ficheiros de entrada).

Os pedidos de contratação devem ir sendo atribuídos a operadores de acordo com a ordem de chegada desses pedidos, seguindo a ordem no ficheiro de entrada com a listagem de pedidos.

No ficheiro de saída com a calendarização dos pedidos atribuídos, estes são ordenados pela hora crescente de atendimento, do início para o fim do ficheiro (e depois por ordem alfabética dos nomes dos clientes, em caso de empate).

O **especialista** a quem é atribuído o atendimento tem de ser um profissional no domínio do pedido, com um nível de reputação igual ou superior ao pretendido pelo cliente, a cobrar à hora, no máximo, aquilo que o cliente aceita pagar, e a trabalhar na zona/cidade para que é feito o pedido. Uma vez satisfeitos estes requisitos, o especialista a quem é atribuído o atendimento é aquele que estiver mais cedo disponível; em caso de empate, o que cobrar menos à hora; em caso de empate, o que tiver menos montante de remunerações acumulado; em caso de empate, o primeiro por ordem alfabética dos nomes dos especialistas.

A um mesmo especialista será atribuído mais de um pedido se após a atribuição de pedidos anteriores, esse especialista é o que melhor satisfaz as condições de atribuição de pedidos seguintes.

Caso não haja nenhum especialista que satisfaça as condições, deve ser colocada uma linha para o pedido a atender com a data do cabeçalho e o nome do utilizador e a expressão reservada `declined`, como neste exemplo:

```
2018-10-30, 17:00, Jean Michaux, declined
```

Havendo mais de uma linha com a indicação `declined`, tais linhas devem ser as primeiras da listagem e devem ser ordenadas de cima para baixo segundo a ordem lexicográfica dos nomes de clientes que lá constarem.

- Assume-se que a **atualização** da listagem dos especialistas e a **calendarização** do atendimento dos pedidos de contratação são feitas de trinta em trinta minutos.
Em cada dia, o período de funcionamento é das 8h00 às 20h00.
Tarefa cujo início tenha de passar para o dia seguinte, começa às 8h00 desse dia. Tarefa que não possa ser concluída até às 20h00, recomeça no dia seguinte pelas 8h00 e dura o tempo que restar.
Em vista de conter o projeto dentro dos seus limites pedagógicos, para simplificação, assume-se que: um especialista toma 1 hora para mudar de uma tarefa para a próxima; todos os meses do ano têm 30 dias; os especialistas trabalham todos os dias do ano.
Assume-se também que a listagem dos pedidos de ajuda/contratação é feita fora do software que estamos a desenvolver neste projeto.

Especificação em pormenor

A especificação em pormenor do programa é feita através da especificação das suas funções, de acordo com as convenções adotadas no curso. As especificações nos esqueletos dos módulos fornecidos juntamente com este enunciado não se destinam a ser exemplificativas: têm de ser seguidas.

Programação por contrato tem de ser a abordagem seguida.

Estrutura da aplicação

A aplicação `cyberConcierge` é composta pelo programa

```
cyberConc.py
```

e ainda pelos seguintes módulos a que este recorre:

```
constants.py  
dateTime.py  
filesReading.py  
scheduling.py  
filesWriting.py
```

Estes módulos devem incluir, entre possivelmente outras funções que entender necessárias ou convenientes, as funções apresentadas nos esqueletos e nos stubs disponibilizados em associação com o presente enunciado.

O código desses módulos e funções tem de ser completado e pode ter de ser corrigido.

As especificações fornecidas têm de ser respeitadas e as restantes têm de ser completadas.

O programa `cyberConc.py` por sua vez contém uma função cuja chamada assegura o funcionamento da aplicação.

Exceções

A aplicação deve lançar a exceção:

```
Error in input file: inconsistent name and header in file <name of  
file>.
```

quando num ficheiro de entrada se verificar inconsistência entre o seu nome e o seu cabeçalho quanto à data (e.g. 2018-10-30) ou à hora (e.g. 01h30) e/ou ao âmbito (e.g. Experts, Clients ou Schedule).

Deve lançar a exceção:

```
Error in input files: inconsistent files <name of file1> and <name of file2>.
```

quando entre dois ficheiros de entrada se verificar inconsistência entre os seus cabeçalhos (exceptuando a última linha do cabeçalho, relativa ao âmbito).

De forma a conter o projeto dentro dos seus limites pedagógicos, as pré-condições sobre a restante estrutura interna dos ficheiros de entrada, respeitante ao formato de arrumação da informação, exemplificado acima, não devem ser verificadas (assumimos que os ficheiros de entrada vêm todos bem estruturados).

Linguagem

A linguagem de entrada e saída do software para utilizadores humanos é o inglês.

A linguagem da documentação, especificação, nomeação de funções, variáveis e constantes, comentários no código etc é também o inglês.

Executar o software

O software é executado através da seguinte instrução na linha de comandos:

```
python cyberConc.py inputFile1 inputFile2
```

`inputFile1` é um ficheiro com a listagem dos **especialistas**, `inputFile2` é um ficheiro com a listagem dos **pedidos de contratação**, indicado na linha de comando por essa ordem.

Os ficheiros de saída produzidos são escritos na mesma diretoria onde se encontram os ficheiros de entrada. Um, com a calendarização, tem o nome `YYYYyMMmDDscheduleXXhYY.txt`, e o outro, com a listagem atualizada dos especialistas, o nome de `YYYYyMMmDDexpertsXXhYY.txt`, em que `YYYYyMMmDD` e `XXhYY` devem representar a data e a hora que resulta de acrescentar trinta minutos ao tempo e hora indicados nos ficheiros de entrada, levando em consideração o horário de funcionamento das 8h00 às 20h00.

Dicas

Para ordenação de coleções, sugere-se a utilização do método `sort` ou da função `sorted` da biblioteca padrão do Python. Um pequeno manual encontra-se aqui: <https://wiki.python.org/moin/HowTo/Sorting>

Para obter os nomes dos ficheiros a partir da instrução de arranque do programa na linha de comandos acima indicada, sugere-se a utilização da variável `argv` do módulo `sys`. Especificação e explicação encontram-se aqui:

<https://docs.python.org/3/library/sys.html>

https://www.tutorialspoint.com/python/python_command_line_arguments.htm

2. Desenvolvimento do software

Grupos

O projeto tem de ser realizado por grupos de exatamente 2 alunos. Cada estudante ERASMUS deve fazer grupo com um estudante não-ERASMUS. Os grupos podem conter alunos de diferentes turmas. Os grupos registam-se no site da disciplina.

A única FORMA DE REGISTO de grupos é através do site da disciplina, em:
<https://moodle.ciencias.ulisboa.pt/course/view.php?id=1156>

Elementos fornecidos aos alunos

Para a elaboração da componente de avaliação respeitante ao projeto, são fornecidos os seguintes elementos, que se encontram no site da disciplina:

- presente enunciado
- esqueleto dos módulos com especificações das funções
- exemplos com os ficheiros de entrada e correspondentes ficheiros de saída

Máximas

Os estudantes a realizar o presente projeto são tipicamente programadores principiantes. Têm toda a vantagem em observar as seguintes máximas, que ainda não tiveram oportunidade de descobrir/consolidar por si próprios:

1. "já"

positivo: começar a resolver o projeto agora, no momento em que este enunciado foi publicado

negativo: esperar até alguns dias antes do prazo de entrega para começar leva ao desastre

2. "passo a passo"

positivo: ir fazendo e testando pequenas partes do código progressivamente

negativo: esperar para testar até haver uma primeira versão total ou completa leva ao desastre

3. "desbloquear rápido"

positivo: falar com os docentes (e colegas) para esclarecer dúvidas e desbloquear impasses logo que estes surgem

negativo: esperar por futuro rasgo solitário de inspiração súbita leva ao desastre

Apoio para a resolução do projeto

Continuam ao dispor os meios de apoio pedagógico para os alunos desta disciplina, que se encontram disponíveis desde o início do curso, e que podem e devem ser usados para apoio à resolução do presente projeto. Relembra-se que são os seguintes:

- contato com os docentes ao **final das aulas** ao longo do semestre
- horários de **atendimento** presencial, individual e personalizado, aos alunos ao longo da semana
<https://moodle.ciencias.ulisboa.pt/course/view.php?id=1156>
- **forum de entreajuda** da disciplina, com acesso por todos os estudantes
<https://moodle.ciencias.ulisboa.pt/mod/forum/view.php?id=56918>
- espaço de **notícias** da disciplina
<https://moodle.ciencias.ulisboa.pt/mod/forum/view.php?id=48991>

Dada a natureza da tarefa a concretizar e o contexto do código em que eventuais dificuldades surgem, esclarecimentos sobre a resolução do projeto devem ser obtidos através destes meios de apoio, não sendo atendíveis através de mensagens de email para os docentes.

3. A componente de avaliação

Elementos a entregar pelos alunos para avaliação

Uma pasta com o ficheiro com o relatório de implementação e com os ficheiros de código desenvolvido, incluindo os seguinte seis ficheiros (e outros se for o caso):

```
cyberConc.py  
constants.py  
dateTime.py  
filesReading.py  
filesWriting.py  
scheduling.py
```

A pasta deve ter o nome `cyberConcGroupN`, em que `N` é o número do grupo, atribuído no processo de inscrição do grupo. Por exemplo, para o grupo de alunos que recebeu o número 2034, a pasta deve ter o nome `cyberConcGroup2034`.

A pasta tem de ser submetida zipada, com o nome `cyberConc2034.zip`.

Cada um dos ficheiros de código, por sua vez, tem de conter nas primeiras linhas, como comentários, informação sobre o número do grupo e número e nome completo de cada membro do grupo que trabalhou no projeto, como exemplificado a seguir:

```
#2018-2019 Programação 1 (LTI)  
#Grupo 2034  
#55123 Florentino Ariza  
#55456 Blimunda Tortesindo
```

Ficheiros de código sem algum destes elementos não serão avaliados.

Relatório de implementação

O relatório de implementação não deve ultrapassar duas páginas, tem o nome `relGrupoN.pdf` (em que `N` é o número do grupo) e tem de estar no formato `.pdf` (relatórios noutros formatos serão ignorados). Tem de ser estruturado de acordo com as seguintes **secções**:

1. Número do grupo
2. Número e nome completo de cada membro do grupo
3. Indicação detalhada do que cada membro do grupo fez para a resolução do projeto
4. Indicação de funções extra implementadas (se aplicável) e do seu funcionamento
5. Indicação das funcionalidades que ficaram por implementar (se aplicável)
6. Indicação de erros conhecidos (se aplicável)

O relatório pode ser escrito em português ou em inglês.

Dimensões em avaliação

Os projetos serão avaliados de acordo com as seguintes dimensões e ponderações:

- A. 1 se está completo e funciona sem gerar erros ao compilar e correr sobre exemplos (1, 2 e 3) fornecidos, 0 caso contrário
- B. Correção semântica (funciona como especificado no enunciado), 60%
- C. Correção pragmática (organizado como indicado no enunciado, estruturas de dados e abordagens algorítmicas ponderadas e práticas de programação apropriadas), 20%
- D. Documentação (especificação, comentários q.b.), 10%
- E. Legibilidade (nomeação perspicua, arrumação e formatação do código), 5%
- F. Relatório de implementação, 5%

A classificação é encontrada através da fórmula $A * (B + C + D + E + F)$

Integridade académica

Como futuro profissional, espera-se de si uma atitude irrepreensível em termos éticos e deontológicos. Tenha pois o maior cuidado em respeitar e fazer respeitar a lei da criminalidade informática.

Alunos detetados em situação de fraude ou plágio parcial ou total - plagiadores e plagiados, com ou sem a intervenção de intermediários - em alguma componente de avaliação ficam liminarmente com esta prova cancelada e serão alvo de processo disciplinar, o que levará a um registo dessa incidência no processo de aluno. Não queira ter de mostrar o seu diploma a um futuro empregador com uma incidência dessas registada.

Pode e deve haver entreaajuda entre alunos, através da discussão de métodos e algoritmos aplicáveis. É porém da exclusiva responsabilidade de cada grupo tomar medidas para proteger o seu código de ser plagiado.

No processo de avaliação será usado software de apoio na detecção de plágio que compara a resposta de cada grupo com cada uma das respostas dos outros grupos.

Forma e data de entrega

Para submeterem a solução do vosso grupo a avaliação, **entregam um FICHEIRO .zip**, que resulta de se comprimir a pasta com os ficheiros de código desenvolvidos e o relatório (por exemplo, cyberConcGroup2034.zip).

A **ÚNICA FORMA DE ENTREGA é a através do site da disciplina, em:**
<https://moodle.ciencias.ulisboa.pt/course/view.php?id=1156>

Qualquer entrega noutra forma não será considerada para avaliação.

Para ser avaliada, a vossa solução deve ser submetida até ao **PRAZO de sexta-feira, 14 de Dezembro de 2018, 23h00 (hora de Lisboa).**

Qualquer entrega ou resubmissão depois deste prazo não será considerada para avaliação.