

Fundamentos de Programação, 2017/2018

Projeto

(este enunciado tem 13 páginas)

byChatWeHelp

0. Contexto

No mercado dos dispositivos tecnológicos e dos serviços remotos avançados, um apoio de qualidade aos utilizadores na resolução de dificuldades com a instalação, operação e rentabilização dos mesmos constitui um fator decisivo de sucesso para os agentes económicos neste sector.

Neste enquadramento, os centros de atendimento ("call centers") têm sido objeto de uma evolução constante ao longo dos últimos anos. Assistimos neste contexto a um crescimento explosivo de uma nova modalidade de contacto, que vem acompanhar a via telefónica tradicional, nomeadamente o contacto feito através de web chat, correio electrónico, media sociais, etc. Estudos de mercado indicam que, dentro de pouco tempo, este tipo de contacto vai ultrapassar o contacto baseado no telefone.



Imagem de um centro de contacto.

Nesta medida, é interessante notar que, por exemplo, o contacto através de chat permite a redução de custos por via da automatização das respostas aos utilizadores. Verifica-se que, dentro de um dado domínio (e.g. impressoras, telemóveis, televisores, etc) a esmagadora maioria de pedidos de apoio se refere a um número muito restrito de problemas tipo a resolver. Deste modo, as respostas a estes pedidos podem constar de uma base de dados e serem automaticamente entregues de cada vez que um desses pedidos de ajuda muito frequentes entrar.

Sobram porém os pedidos de ajuda inéditos ou muito raros. Estes terão de ser reencaminhados para um dos operadores humanos no centro de contacto.

A NSHF (Not So High Functions) é uma empresa start up que pretende explorar as oportunidades de negócio oferecidas por esta crescente automatização das respostas aos pedidos de ajuda remotos pelos utilizadores. Parte da sua operação vai assentar no direccionamento para operadores humanos dos pedidos de ajuda por chat que não poderem ser respondidos automaticamente.

Para tanto precisará de recorrer a um software a desenvolver, batizado de byChatWeHelp, que assegure a gestão dos pedidos de ajuda, dos operadores humanos e do emparelhamento entre uns e outros de forma a minimizar os custos e a maximizar a qualidade do serviço.

Por exemplo, um pedido de ajuda sobre telemóveis deve ser direccionado para um operador especializado nesta área, e não na área de televisores; um pedido por um cliente premium deve ter prioridade sobre um pedido de um cliente freemium, 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 3.2, neste projeto vai implementar o software `byChatWeHelp`. É um software que apoia o direcionamento de pedidos de ajuda usado pela empresa NSHF para gerir a atribuição de pedidos de ajuda por chat aos operadores do seu centro de contacto.

Funcionalidade

O seu programa recebe uma listagem dos operadores a qual caracteriza, num dado momento, cada um dos operadores quantos a aspetos relevantes para o seu atendimento de pedidos de ajuda dos utilizadores. O seu programa recebe também uma listagem dos pedidos de ajuda que se encontram por atribuir a operadores até esse mesmo dado momento.

Por um lado, o seu programa entrega um plano de atendimento desses pedidos de ajuda pelos operadores.

Por outro lado, entrega ainda a listagem atualizada dos operadores, 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:

`operators14h55.txt`

```
Day:
06:11:2017
Time:
14:55
Company:
NSHF
Operators:
Ricardo Tavares, portuguese, (mobiles; printers), 14:15, 42
Giovanni Olivetti, italian, (laptops; bimby; hifi), 14:52, 21
Georg Muller, deutsch, (cameras), 15:05, 31
...
David Yard, ...
```

requests14h55.txt

```
Day:
06:11:2017
Time:
14:55
Company:
NSHF
Requests:
Henry Miller, english, laptops, fremium, 3
François Greenwich, spanish, cameras, premium, 5
Ricardo Carvalho, portuguese, refrigerators, premium, 1
...
```

Os pedidos estão ordenados, de cima para baixo, do mais antigo ao mais recente, por atender.

Saída

O programa produz dois ficheiros, um com a listagem dos operadores atualizada e outro com a calendarização da execução dos pedidos de ajuda. Com uma finalidade pedagógica, assume-se que as atualizações e a calendarizações são feitas em simultâneo e de cinco em cinco minutos.

O ficheiro com a listagem de operadores atualizada tem uma estrutura interna similar ao ficheiro de entrada com a listagem dos operadores. A diferença é que o cabeçalho é atualizado quanto à data e ao tempo (incrementado de 5 minutos em relação ao momento do ficheiro de entrada), e os dois últimos campos de cada operador, um com o tempo acumulado de atendimento e o outro com a hora em que ficará liberto para atender novo pedido não calendarizado, são atualizados em função da calendarização feita dos pedidos de ajuda.

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

timetable15h00.txt

```
Day:
06:11:2017
Time:
15:00
Company:
NSHF
Timetable:
16:56, François Greenwich, Ramon Diaz
16:56, Henry Miller, Elizabeth Conrad
16:58, Ricardo Carvalho, José Soares
...
```

Mais sobre especificação geral

- As diferentes listagens (operadores, pedidos, calendarização) são guardadas em ficheiros .txt.
- Cada listagem começa com um cabeçalho que contém a indicação do dia da operação, do tempo da operação, da companhia e do âmbito do ficheiro (Operators, Requests ou Timetable) como neste exemplo:

```
Day:  
17:10:2017  
Time:  
10:25  
Company:  
Whizbang  
Timetable:
```

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

```
timetable10h25.txt
```

- Na **listagem de operadores**, a seguir ao cabeçalho, cada linha corresponde a um operador (cujos respetivos elementos informativos estão separados por vírgulas) estando a listagem ordenada de cima para baixo por ordem decrescente de disponibilidade do operador (e depois por ordem alfabética dos nomes dos operadores em caso de empate).
Cada operador é caracterizado por nome (e.g. Gustaf Svenson), língua de atendimento (e.g. swedish), domínios da sua especialidade, entre parêntesis e separados por ponto e vírgula (e.g. (mobiles; printers)), hora de finalização do último pedido que vai atender (e.g. 14:15) (n.b.: para atribuição de nova tarefa ao operador, contará a mais tardia de entre esta data e a data do cabeçalho), e número de minutos de atendimento já realizados (incluindo todos os que estão calendarizados), como ilustrado no seguinte exemplo:

```
Gustaf Svenson, swedish, (mobiles; printers), 14:15, 42
```

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

- Na **listagem de pedidos de ajuda**, a seguir ao cabeçalho, cada linha corresponde a um pedido de ajuda (cujos elementos informativos estão separados por vírgulas) estando a listagem ordenada, de cima para baixo, pela ordem de chegada dos pedidos.
Cada pedido de ajuda é caracterizado por um utilizador (e.g. François Greenwich), pelo idioma em que o pedido se encontra escrito (e.g. spanish), o domínio do pedido (e.g. cameras), o regime de utilização

(premium, se o utilizador paga subscrição do serviço; fremium se não paga qualquer subscrição), e a duração do período que vai levar a responder pelo operador, em minutos (e.g. 5), como ilustrado no seguinte exemplo:

François Greenwich, spanish, cameras, premium, 5

A duração do período que vai levar a responder é estimado automaticamente por uma outra aplicação com base na pergunta, no domínio, história prévia de pedidos do mesmo utilizador, etc. Para efeitos pedagógicos vamos considerar esta duração como certa e fixa.

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 dos operadores.

- Na **calendarização do atendimento dos pedidos de ajuda**, 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 hora de início (e.g. 16:56), pelo nome do utilizador que fez o pedido (e.g. Henry Miller), e pelo nome do operador que vai fazer esse atendimento (e.g. Elizabeth Conrad), como ilustrado no seguinte exemplo:

16:56, Henry Miller, Elizabeth Conrad

A hora de início é a hora mais tardia entre a hora de início do período e a hora de disponibilidade do operador.

O cabeçalho é similar aos dos ficheiros de entrada, atualizado quanto ao tempo (incrementado de 5 minutos em relação ao momento dos ficheiros de entrada).

Os pedidos de apoio devem ir sendo atribuídos a operadores de acordo com a sua ordem de chegada, 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 por depois por ordem alfabética dos nomes dos clientes, em caso de empate).

O **operador** a quem é atribuído o atendimento tem de utilizar o mesmo idioma do pedido e tem de ser proficiente no domínio do pedido.

Uma vez satisfeitos estes requisitos, o operador a quem é atribuído o atendimento é aquele que estiver mais cedo disponível, em caso de empate, o que tiver menos tempo acumulado, em caso de empate, por ordem alfabética dos nomes dos operadores, desde que essa atribuição não ultrapasse o tempo máximo de atendimento acumulado por dia que esse operador pode realizar (fixado em 240 minutos):

- Se o pedido for no regime premium, deve ser atribuído ao operador que mais cedo esteja disponível.

- Se o pedido for no regime *fremium*, deve ser atribuído ao operador que mais cedo esteja disponível depois de todos os pedidos *premium* terem sido atribuídos.

A um mesmo operador será atribuído mais de um atendimento se após a atribuição de atendimentos anteriores, esse operador é o que melhor satisfaz as condições do pedido em causa.

Caso não haja nenhum operador 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 *not-assigned*, como neste exemplo:

16:55, Liu Xiau, not-assigned

- Assume-se que a **atualização** da listagem dos operadores e a **calendarização** das tarefas de tradução são feitas em simultâneo, de cinco em cinco minutos.

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 *byChatWeHelp* é composta pelo programa

`helpchat.py`

e ainda pelos seguintes módulos a que este recorre:

`constants.py`
`dateTime.py`
`filesReading.py`
`assigning.py`
`filesWriting.py`

Estes módulos devem incluir, entre possivelmente outras funções que entender necessárias ou convenientes, as funções apresentadas nos esqueletos e 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 `helpchat.py` por sua vez contém uma função cuja chamada assegura o funcionamento da aplicação.

Exceção

Deve ser lançada a exceção:

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

quando num ficheiro de input se verificar inconsistência entre o seu nome e o seu cabeçalho quanto ao tempo (e.g. 01h35) e/ou ao âmbito (e.g. operators, requests ou timetable).

Deve ser lançada a exceção:

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

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

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

Linguagem

A linguagem do input e output do software para utilizadores humanos é o inglês.

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

Executar o software

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

```
python helpchat.py inputFile1 inputFile2
```

`inputFile1` é um ficheiro com a listagem dos **operadores**, `inputFile2` é um ficheiro com a listagem dos **pedidos de ajuda**, indicado na linha de comando por esta ordem.

Os ficheiros de saída produzidos são escritos na mesma diretoria onde se encontram os ficheiros de input. Um, com a calendarização, tem o nome `timetableXXhYY.txt`, e o outro, com a listagem atualizada dos operadores, o nome de `operatorsXXhYY`, em que XXhYY deve representar a hora que resulta de acrescentar 5 minutos ao tempo e data indicados nos ficheiros de entrada.

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/2/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=4>

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 input e correspondentes ficheiros de output

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 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. Lembra-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=4>
- **fórum de entreaajuda** da disciplina, com acesso por todos os estudantes
<https://moodle.ciencias.ulisboa.pt/mod/chat/view.php?id=34954>
- espaço de **notícias** da disciplina
<https://moodle.ciencias.ulisboa.pt/mod/forum/view.php?id=33457>

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 desenvolvidos, incluindo os seguinte sete ficheiros (e outros se for o caso):

```
assigning.py  
constants.py  
dateTime.py  
filesReading.py  
filesWriting.py  
helpchat.py
```

A pasta deve ter o nome `chatHelpGroupN`, 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 943, a pasta deve ter o nome `chatHelpGroup943`.

A pasta tem de ser submetida zipada, com o nome `chatHelp943.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:

```
#2017-2018 Programação 1 (LTI)  
#Grupo 943  
#55123 Belarmina Crisálida  
#55456 Idalécio Vicentino
```

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 entreajuda 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, chatHelpGroup943.zip).

A ÚNICA FORMA DE ENTREGA é a através do site da disciplina, em:

<https://moodle.ciencias.ulisboa.pt/course/view.php?id=4>

Qualquer entrega noutra forma não será considerada para avaliação.

Para ser avaliada, a vossa solução deve ser submetida até ao **PRAZO de quarta-feira, 13 de Dezembro de 2017, 23h00 (hora de Lisboa)**.

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