

Projeto

(este enunciado tem 13 páginas)

dronyDeliv

0. Contexto

Baterias mais leves e maior capacidade de acumulação de energia, novos materiais mais resistentes, motores mais silenciosos e potentes, redes de comunicação com maior largura de banda, etc estão a criar novas oportunidades de negócio na entrega automática de encomendas porta-a-porta com drones.



Fartos de serem explorados em tarefas de entrega da economia-gig pagas miseravelmente e sem condições de trabalho minimamente aceitáveis, dois amigos, Francisco e José, decidiram associar-se para fundarem uma startup que disponibiliza serviços de entrega automática de encomendas. O trabalho com os equipamentos e infraestruturas da empresa está já muito avançado: um grande armazém foi alugado, centenas de drones foram adquiridos e vêm a caminho, a estação de reabastecimento está pronta, a oficina de drones está quase montada, os profissionais para as diferentes funções começaram a ser contratados, etc.

Tudo parecia muito bem encaminhado para que a empresa começasse a sua atividade em breve quando os dois sócios, com alarme, se deram conta de uma lacuna no seu planeamento: como decidir que drone realiza que entrega? Deram-se conta de que estava a faltar uma aplicação informática de agendamento de entregas.

Como tinham vários amigos a estudar na Licenciatura em Tecnologias de Informação de Ciências ULisboa, compreenderam que pedir-lhes ajuda seria a única saída com probabilidade de funcionar em pouco tempo e de os tirar da situação desesperada em que se encontravam, sem declararem falência antes de iniciarem o negócio. 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 `dronyDeliv`, usado pela empresa `iXicoZe` para gerir a atribuição de pedidos de transporte de encomendas, feitos por clientes, aos drones.

Funcionalidade

O seu programa **recebe** uma listagem de drones que caracteriza, num dado momento, cada um dos drones quanto a aspetos relevantes para a realização de pedidos de transporte feitos por clientes. Recebe também uma listagem dos pedidos de transporte que se encontram por atribuir a drones até esse mesmo momento.

O seu programa **entrega**, por um lado, um plano de entregas pelos drones das encomendas constantes dos pedidos de transporte dos clientes. Por outro lado, entrega ainda a listagem atualizada dos drones, após os pedidos de transporte terem sido realizados por estes.

Entrada

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

Exemplo 1

`drones15h30_2019y11m4.txt`

```
Time:
15h30
Day:
4-11-2019
Company:
iXicoZe
Drones:
d14, alfama, 15, 1700, 789.0, 123.0, 2019-11-04, 15:17
dGig, madragoa, 20, 1500, 516.0, 230.0, 2019-11-04, 15:27
dXX, santos, 10, 2000, 1003.0, 634.0, 2019-11-04, 15:31
d204, santos, 25, 1200, 908.0, 511.0, 2019-11-04, 15:31
...
```

Na **listagem de drones**, a seguir ao cabeçalho, cada linha corresponde a um drone (cujos respetivos elementos informativos estão separados por vírgulas).

Os drones estão ordenados, de cima para baixo, desde o que estará há mais tempo disponível para o que estará há menos tempo; em caso de empate, do que lhe sobra mais tempo de autonomia para o que lhe sobra menos; e em caso de empate, por ordem lexicográfica crescente do nome do drone.

Cada drone é caracterizado por uma única linha com:

- nome (e.g. d14),
- zona de operação (e.g. alfama),
- peso que pode transportar, em kilogramas (e.g. 15),
- distância máxima da base a que pode voar, em metros (e.g. 1700)
- distância acumulada percorrida até então, em quilómetros (e.g. 789),
- autonomia, em quilómetros (e.g. 123),
- data de disponibilidade (e.g. 2019-11-04), que é a data da finalização do último pedido com a aterragem de volta na base
- hora de disponibilidade (e.g. 15:17), que é a hora da finalização do último pedido com a aterragem de volta na base

como no seguinte exemplo:

```
d14, alfama, 15, 1700, 789, 123, 15:17
```

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

Exemplo 2

parcels15h30_2019y11m4.txt

```
Time:
15h30
Day:
4-11-2019
Company:
iXicoZe
Parcels:
Ana Silva, telheiras, 2019-11-04, 15:30, 1234, 23, 45
Eduardo Nunes, chiado, 2019-11-04, 15:45, 989, 17, 37
Gisela Santos, madragoa, 2019-11-05, 10:00, 1760, 42, 74
...
```

Na **listagem de encomendas em espera**, a seguir ao cabeçalho, cada linha corresponde a um pedido de transporte (cujos elementos informativos estão separados por vírgulas). Os pedidos estão ordenados, de cima para baixo, desde a que tem de sair mais cedo para a que tem de sair mais tarde; em caso de empate, da mais leve para a mais pesada; em caso de empate ainda por ordem lexicográfica crescente do nome do cliente.

Cada encomenda é caracterizada por uma única linha com:

- o nome do cliente que fez o pedido (e.g. Ana Silva),
- a zona para onde o pedido é feito (e.g. telheiras),
- a data a partir da qual o pedido deve ser executado (e.g. 2019-11-04) e a hora (e.g. 15:30),
- a distância do local de entrega em relação à base dos drones naquela zona da cidade, em metros (e.g. 1234),
- o peso da encomenda a transportar, em kilogramas (e.g. 23),

- o o tempo necessário para o transporte, ida e volta, em minutos, até à aterragem de volta na base (e.g. 45),

como no seguinte exemplo:

Ana Silva, telheiras, 2019-11-04, 15:30, 1234, 23, 45

O cabeçalho indica a data e a hora da recolha dos pedidos, que devem ser idênticas à data e hora da última atualização do ficheiro de entrada dos drones.

Saída

O programa produz dois ficheiros, um com a listagem dos drones atualizada e outro com a calendarização do transporte das encomendas. 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 drones atualizada tem uma estrutura interna similar ao ficheiro de entrada com a listagem dos drones, em que cada drone 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 — respetivamente com distância percorrida, autonomia e hora de disponibilidade — são atualizados em função da calendarização feita para os pedidos de transporte.

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

timetable15h30_2019y11m4.txt

```
Time:
15h30
Day:
4-11-2019
Company:
iXicoZe
Timeline:
2019-11-04, 15:30, Ana Silva, dEagle
2019-10-04, 15:45, Eduardo Nunes, d402
...
```

Na **calendarização do atendimento dos pedidos de transporte**, 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; e por ordem lexicográfica crescente do nome do cliente em caso de empate.

Cada transporte calendarizado é caracterizado pela data (e.g. 2019-11-04) e hora de início (e.g. 15:30), pelo nome do cliente que fez o pedido (e.g. Ana Silva), e

pelo nome do drone que vai responder a esse pedido (e.g. dEagle), como ilustrado no seguinte exemplo:

2019-11-04, 15:30, Ana Silva, dEagle

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

O cabeçalho é similar aos dos ficheiros de entrada.

Os pedidos de transporte devem ir sendo atribuídos a drones de acordo com a ordem de saída pedida para esses transportes, seguindo a ordem no ficheiro de entrada com a listagem de pedidos.

No ficheiro de saída com a calendarização dos transportes atribuídas, estas são ordenados pela hora crescente de atendimento, do início para o fim do ficheiro; e depois por ordem lexicográfica dos nomes dos clientes, em caso de empate.

O **drone** a que é atribuído o transporte tem de ser um drone na zona do pedido, poder levar o peso dessa encomenda e com autonomia suficiente para entregar a encomenda e voltar à base. Uma vez satisfeito este requisito, o drone a que é atribuído a transporte é aquele que estiver mais cedo disponível; em caso de empate, com maior autonomia; em caso de empate, com menor distância percorrida acumulada; em caso de empate, o primeiro por ordem lexicográfica crescente dos nomes dos drones.

A um mesmo drone será atribuído mais de um transporte se após a atribuição de transportes anteriores, esse drone é o que melhor satisfaz as condições de atribuição de transportes seguintes.

Caso não haja nenhum drone que satisfaça as condições, deve ser colocada uma linha para o transporte pedido com a data do cabeçalho e o nome do cliente e a expressão reservada `cancelled`, como neste exemplo:

2019-11-30, 17:00, Carlos Soares, cancelled

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

Mais sobre especificação

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

Time:
15:30
Day:
2019-11-04
Company:
iXicoZe
Drones:

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

timetable15h30_2019y11m4.txt

- Assume-se que a **atualização** da listagem dos drones e a **calendarização** do atendimento dos pedidos de transporte são feitas de trinta em trinta minutos.

Em cada dia, o período de funcionamento é das 8h00 às 20h00. Transporte cujo início tenha de passar para o dia seguinte, começa às 8h00 desse dia. Tarefa que não possa ser concluída até às 20h00, começa no dia seguinte pelas 8h00.

Em vista de conter o projeto dentro dos seus limites pedagógicos, para simplificação, assume-se que: todos os meses do ano têm 30 dias; os drones trabalham todos os dias do ano.

Assume-se também que a listagem dos pedidos de transporte é feita fora do software que estamos a desenvolver neste projeto.

- Resultados de cálculos em números decimais são representados por floats arredondados a uma casa decimal.

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 `dronyDeliv` é composta pelo programa

`dronyD.py`

e ainda pelos seguintes módulos a que este recorre:

```
constants.py  
time.py  
readFiles.py  
organize.py  
writeFiles.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 `dronyD.py` por sua vez contém uma função que se deve chamar `allocate`, a ser definida pelo programador, cuja chamada assegura o funcionamento da aplicação.

Exceções

A aplicação deve lançar a exceção:

```
Input error: name and header inconsistent 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. 2019-11-30) ou à hora (e.g. 01h30) e/ou ao âmbito (e.g. Drones, Parcels ou Timeline).

Deve lançar a exceção:

```
Input error: 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 dronyD.py inputFile1 inputFile2
```

`inputFile1` é um ficheiro com a listagem dos **drones**, `inputFile2` é um ficheiro com a listagem dos **pedidos de transporte**, 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 `timelineHHhMM_YYYYyMMmDD.txt`, e o outro, com a listagem atualizada dos drones, o nome de `dronesHHhMM_YYYYyMMmDD.txt`, em que `YYYYyMMmDD` e `HHhMM` 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=1762>

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=1762>
- **fórum de entreajuda** da disciplina, com acesso por todos os estudantes
<https://moodle.ciencias.ulisboa.pt/mod/forum/view.php?id=75541>
- espaço de **notícias** da disciplina
<https://moodle.ciencias.ulisboa.pt/mod/forum/view.php?id=75539>

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):

```
dronyD.py  
constants.py  
time.py  
readFiles.py  
writeFiles.py  
organize.py
```

A pasta deve ter o nome `dronyDelivGroupN`, 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 546, a pasta deve ter o nome `dronyDelivGroup546`.

A pasta tem de ser submetida zipada, com o nome `dronyDelivGroup546.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:

```
#2019-2020 Programação 1 (LTI)  
#Grupo 546  
#55123 Florbela Queirós  
#55456 Anastácio Cunha
```

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, dronyDelivGroup546.zip).

A **ÚNICA FORMA DE ENTREGA é a através do site da disciplina, em:**
<https://moodle.ciencias.ulisboa.pt/course/view.php?id=1762>

Qualquer entrega noutra forma não será considerada para avaliação.

Para ser avaliada, a vossa solução deve ser submetida até ao **PRAZO de sexta-feira, 13 de Dezembro de 2019, 23h00 (hora de Lisboa).**

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