

Programação 2 (LTI), 2022/2023

Projeto

(este enunciado tem 14 páginas)

enjoyTagus

(programação orientada a objetos)

0. Contexto

É sabido que depois das suas carreiras desportivas, os atletas de diferentes modalidades tendem a seguir uma atividade profissional relacionada com o seu desporto: no ensino da modalidade, no treino de equipas, no negócio de produtos altamente especializados desse desporto, etc. Isso é particularmente notório no caso da vela desportiva, em que os ex-praticantes formam uma comunidade muito coesa e os seus elementos partilham entre si oportunidades profissionais, tais como transportar barcos à vela a longas distâncias, assegurar serviços marítimo-turísticos, entre vários outros.



Com a explosão do turismo nos últimos anos em Lisboa, a atividade de cruzeiros no Tejo assistiu a um crescimento assinalável, vendo-se por exemplo agora ao final das tardes calmas de Lisboa, dezenas de barcos à vela junto à frente ribeirinha e em especial frente ao Terreiro do Paço, para aquilo que é divulgado como “sunset cruises”.

Nestes cruzeiros, os skippers são maioritariamente ex-velejadores desportivos, e com a procura sempre crescente pelos turistas dos seus serviços — tanto para o período do “sunset” como qualquer outro período do dia —, enfrentam agora um problema associado ao crescimento da sua atividade. Com vista a otimizar as oportunidades de negócio oferecidas pelo número sempre crescente de turistas que os procuram para realizar cruzeiros no Tejo, precisam de criar uma central de reservas.

Para darem resposta a esta necessidade, decidiram recorrer aos alunos de Programação 1 (LTI) da Faculdade de Ciências de Lisboa, os quais têm pela frente como projeto de programação do seu plano de avaliação, desenvolver a base de um software para fazer face a essa necessidade.

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 implementar a aplicação [enjoyTagusOO](#). É um software que apoia o encaminhamento de turistas e que irá ser usado pela cooperativa Tagus Sailing, de atividades marítimo-turísticas, para gerir a atribuição de pedidos de viagens no Rio Tejo aos skippers na sua rede de cooperantes.

Funcionalidade

O seu programa **recebe** uma **listagem dos skippers**. Esta listagem caracteriza, num dado momento, cada um dos skippers quanto aos aspetos relevantes para o seu atendimento de pedidos de cruzeiros em barco à vela no estuário do Tejo. O seu programa recebe também um **calendário de cruzeiros já agendados** e ainda uma **listagem dos cruzeiros pedidos** por clientes que se encontram por atribuir a skippers desde a última calendarização.

O seu programa **entrega**, por um lado, o **calendário de cruzeiros atualizada**. Por outro lado, entrega ainda a **listagem dos skippers atualizada**, após os pedidos de viagens terem sido analisados e eventualmente distribuídos por eles, conforme as disponibilidades dos mesmos e os requisitos dos pedidos feitos pelos clientes.

Entrada

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

skippers14h00.txt

```
Company:
Tagus Sailing
Day:
07:11:2022
Time:
14:00
Skippers:
Chico Silva, (english; portuguese), 3*, 100, speed, 40, 14,
(08:11:2022, 10:00)
Luísinha Costa, (english; portuguese; spanish), 2*, 80, price,
65, 32, (01:12:2022, 12:00)
Vasco Amaral, (english; french; portuguese), 1*, 40, comfort, 20,
5, (20:12:2022, 09:00)
...
Zé Assureira, ...
```

schedule14h00.txt

```
Company:
Tagus Sailing
Day:
07:11:2022
Time:
14:00
Schedule:
07:11:2022, 10:00, 4, Chico Silva, 400, Kadri Vider
07:11:2022, 12:00, 2, Nuno Melo, 120, Ehrard Hinrichs
08:11:2022, 8:00, 1, Chico Silva, 100, Francisca de Jong
...
14:11:2022, 14:00, 4, Mário Santos, 400, Vladislav Maraev
```

requests14h00.txt

```
Company:
Tagus Sailing
Day:
07:11:2020
Time:
14:00
Requests:
Vladislav Abramovich, (english; russian), 2*, price, 10
John Smith, (english), 1*, speed, 2
François Artaud, (french; german), 3*, comfort, 4
Juan Ramirez, (spanish), 2*, speed, 1
...
```

Saída

O programa produz dois ficheiros, um com a listagem dos skippers atualizada e outro com a calendarização dos cruzeiros atualizada. Com uma finalidade pedagógica, assume-se que estas duas atualizações são feitas em simultâneo e de 30 em 30 minutos.

O ficheiro de saída com a listagem de skippers atualizada tem uma estrutura interna similar ao ficheiro de entrada com a listagem dos skippers. A diferença é que o cabeçalho do ficheiro de saída é atualizado quanto ao tempo (incrementado de 30 minutos em relação ao momento do ficheiro de entrada), e quanto a dois campos de cada skipper: um campo com o número de horas acumulado para realizar; e o outro campo com a data e hora de fim do mais tardio dos seus cruzeiros já atribuídos.

O ficheiro com a calendarização dos pedidos de cruzeiro atualizada tem uma estrutura interna para arrumação de informação similar à do respetivo ficheiro de entrada. A diferença para a versão de entrada é que os pedidos de cruzeiros atribuídos a skippers

por via da atualização foram acrescentados e os cruzeiros entretanto já realizados até ao tempo da atualização foram removidos.

Mais sobre especificação geral

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

```
Company:  
Tagus Sailing  
Day:  
07:11:2020  
Time:  
13:00  
Skippers
```

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

```
skippers09h30.txt
```

- Na **listagem de skippers**, a seguir ao cabeçalho, cada linha corresponde a um skipper (cujos respetivos elementos informativos estão separados por vírgulas) estando a listagem ordenada de cima para baixo por ordem alfabética do nome do skipper.

Cada skipper é caracterizado por: nome (e.g. Chico Silva); línguas que domina (e.g. (english; portuguese)) por ordem alfabética; a categoria da sua carta de navegação (de menor para maior categoria: 1*, 2*, ou 3*); tarifa que cobra em euros por hora (e.g. 100); especialização (speed, comfort ou price); o tempo, em número de horas, que aceita navegar sem tirar 2 dias de folga (e.g. 40); número de horas acumulados dos cruzeiros que aceitou fazer e que está a realizar ou ainda vai realizar (e.g. 14); data e hora em que terminará o último cruzeiro que irá realizar (e.g. (08:11:2022, 10:00)) (n.b.: para atribuição de novo cruzeiro ao skipper, contará a mais tardia de entre esta data e hora e a data e hora do cabeçalho), como ilustrado no seguinte exemplo:

```
Chico Silva, (english; portuguese), 3*, 100, speed, 40, 14,  
(8:11:2022, 10:00)
```

Cada cruzeiro dura múltiplos de 30 minutos.

Para cada skipper, o número de horas acumuladas, que está a realizar e que ainda vai realizar, aumenta no momento em que lhe é atribuído um cruzeiro.

O cabeçalho indica a data e a hora da última atualização da listagem dos skippers.

Com uma finalidade pedagógica, assume-se as seguintes simplificações: As atualizações são feitas de 30 em 30 minutos a contar da hora certa; O período de trabalho é das 8h às 20h em todos os dias do ano; Os cruzeiros não param aos feriados ou fins de semana; Todos os meses do ano de todos os anos têm 30 dias. Os períodos de descanso dos skippers e o seu impacto na calendarização dos cruzeiros são tratados por outra aplicação.

- Na **listagem de pedidos de cruzeiros**, a seguir ao cabeçalho, cada linha corresponde a um pedido de cruzeiro (cujos elementos informativos estão separados por vírgulas), estando a listagem ordenada pela ordem de chegada dos pedidos.

Cada pedido de viagem é caracterizado pelo nome do cliente (e.g. Vladislav Abramovich), pelos idiomas que este domina (english; russian), o nível exato de competência do skipper que procura (e.g. 2*), a especialidade do skipper pretendida (e.g. price), e o número de horas para o cruzeiro (e.g. 10), como ilustrado no seguinte exemplo:

Vladislav Abramovich, (english; russian), 2*, price, 10

Cada cruzeiro começa à hora certa ou aos 30 minutos depois da hora certa e dura múltiplos de 30 minutos.

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 dos ficheiros dos skippers e da calendarização.

- Na **calendarização dos cruzeiros**, a seguir ao cabeçalho, cada linha corresponde a um cruzeiro calendarizado (cujos elementos informativos estão separados por vírgulas) estando as linhas na listagem ordenadas por ordem crescente do momento de início de cada cruzeiro (em cada linha). Em caso de linhas com igual momento de início, o desempate faz-se por ordem lexicográfica crescente dos nomes dos skippers.

Cada pedido calendarizado (em cada linha) é caracterizado: pela data e hora a que começa essa viagem (e.g. 07:11:2022, 10:00); pela duração em horas (e.g. 3.5); pelo nome do skipper (e.g. Chico Silva); pelo custo do cruzeiro em euros (e.g. 400); e pelo nome do cliente (e.g. Kadri Vider), como ilustrado no seguinte exemplo:

07:11:2022, 10:00, 3.5, Chico Silva, 350, Kadri Vider

O cabeçalho do ficheiro de saída é similar ao cabeçalho do respetivo ficheiro de entrada, atualizado quanto à data e ao tempo (incrementado de 30 minutos em relação ao momento dos ficheiros de entrada).

Os pedidos de viagens devem ser atribuídos a **skippers** de acordo com a **ordem de chegada** desses pedidos, seguindo a ordem no ficheiro de entrada com a listagem de pedidos.

O **skipper** a quem é atribuído um cruzeiro é aquele que fala alguma das línguas faladas pelo cliente, tem a categoria escolhida pelo cliente e satisfaz a especialidade indicada no pedido. É o skipper que primeiro está disponível para realizar esse cruzeiro. Em caso de empate, é o skipper com a tarifa mais baixa. Em caso de empate ainda, é o que tem menos horas acumuladas. Em caso de empate ainda, é aquele cujo nome é primeiro de acordo com a ordem lexicográfica.

Essa atribuição é feita desde que a realização desse cruzeiro não ultrapasse as 20h do dia respetivo (se ultrapassar, o seu início passa para o dia seguinte) e não ultrapasse o volume máximo de horas acumulado que esse skipper aceita trabalhar sem dias de folga (se ultrapassar, terá de ser atribuído a outro skipper; ou se não houver nenhum outro skipper em condições de o realizar, ficará sem ser atendido e realizado).

Caso não haja nenhum skipper que satisfaça as condições do pedido, no topo da listagem da calendarização deve ser colocada uma linha para a viagem pedida com a respetiva data e o nome do cliente e as palavras reservadas `not-assigned` como neste exemplo:

```
07:11:2022, not-assigned, John Smith
```

Caso exista mais de uma linha deste género, tais linhas ficam ordenadas de acordo com a ordem de chegada dos respetivos pedidos.

O **custo** da realização do cruzeiro é determinado pelo número de horas a velejar e pela tarifa cobrada pelo respetivo skipper.

- No ficheiro de saída com a calendarização dos cruzeiros atribuídos, estes são ordenados pela **data crescente** do início da sua realização, do início para o fim do ficheiro. Em caso de linhas com igual momento de início, o desempate faz-se por ordem lexicográfica crescente dos nomes dos **skippers**.
- Assume-se que a **atualização** da listagem dos **skippers** e da **calendarização** dos cruzeiros são feitas em simultâneo, de 30 em 30 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.

Programação orientada a objetos tem de ser a abordagem seguida.

Programação por contrato tem de ser a abordagem seguida.

As especificações nos esqueletos dos módulos fornecidos juntamente com este enunciado não se destinam a ser exemplificativas: têm de ser seguidas.

Para se libertarem de eventual enviesamento da anterior abordagem não orientada a objetos, que usaram em Programação 1 caso tenham resolvido o respetivo projeto, esses alunos são aconselhados a desenvolver o código da sua resolução do presente projeto a partir da situação de "folha em branco", em vez de alterações incrementais do código do projeto de Programação 1.

Estrutura da aplicação

A aplicação `enjoyTagus00` é composta por classes que o grupo entender como relevantes para a resolução do projeto.

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

Exceção

Deve ser lançada a exceção:

```
Input file error: scope or time 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. 17h30) e/ou ao âmbito (`skippers`, `schedule` ou `requests`). Deve ser lançada uma exceção por qualquer ficheiro de entrada em que ocorrer algum deste tipo de inconsistência.

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, ou seja: 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. A chamada notação camelo (*camelCase notation*) deve ser seguida.

Executar o software

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

```
python3 update.py inputFile1 inputFile2 inputFile3
```

ATENÇÃO:

inputFile1 é um ficheiro com a listagem dos **skippers**.

inputFile2 é um ficheiro com a **calendarização dos cruzeiros**.

inputFile3 é um ficheiro com a listagem dos **pedidos de cruzeiros**.

A ordem destes ficheiros na linha de comandos tem de ser a indicada acima.

Os ficheiros de saída produzidos pela aplicação são escritos na mesma diretoria onde se encontram os ficheiros de input e os de código. Um, com a calendarização, tem o nome `scheduleXXhYY.txt`, e o outro, com a listagem atualizada dos `skippers`, o nome de `skippersXXhYY`, em que `XXhYY` deve representar a hora que resulta de acrescentar 30 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/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. [Dois alunos que formaram grupo no semestre passado para o projeto de Programação 1 não podem formar grupo para este projeto.](#) 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 moodle da disciplina.

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
- exemplos de teste

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 a resolver 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 (indicado no Guião da disciplina, na respetiva página moodle)
- **fórum de entreajuda** da disciplina, com acesso por todos os estudantes (na página moodle da disciplina)
- espaço de **notícias** da disciplina (na página moodle da disciplina)

Dada a natureza da presente 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 indicados acima, não sendo atendíveis através de mensagens de email para os docentes.

3. A componente de avaliação

Elementos a entregar pelos alunos para avaliação

A entregar: uma pasta com o ficheiro com o **relatório** de implementação, com os ficheiros de **código** desenvolvidos e com os ficheiros de **teste** usados.

A pasta deve ter o nome `enjoyTagusGroupN`, 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 `enjoyTagusGroup546`. A pasta tem de ser submetida **zipada**, com o nome `enjoyTagusGroup546.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:

```
#2022-2023 Programação 2
#Grupo 546
#58123 Felismina Felicidade Felisbela
#57456 Anacleto da Costa Anaclista
```

Ficheiros de código sem algum destes elementos não serão avaliados.

Relatório de implementação

O relatório de implementação, com um **máximo de 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 genérica das classes implementadas, dos seus métodos e seu funcionamento
5. Indicação das funcionalidades da aplicação 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

Cada resolução submetida para avaliação será avaliada de acordo com as seguintes dimensões e ponderações:

- A. 1 se está completa e funciona sem gerar erros ao compilar e correr sobre todos os exemplos de teste fornecidos, 0 caso contrário
- B. Correção pragmática (organizado segundo a abordagem OO indicada no enunciado, estruturas de dados e abordagens algorítmicas ponderadas e práticas de programação apropriadas), 60%
- C. Correção semântica (funciona como especificado no enunciado), 20%
- D. Documentação (especificação das funções completas, comentários q.b.), 10%
- E. Legibilidade (nomeação de variáveis e funções perspicua, arrumação e formatação do código, espaçamento apropriado e consistente, sem *scroll* horizontal para ser lido, etc), 5%
- F. Relatório de implementação, 5%

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

Integridade académica

A resolução submetida para avaliação tem de ter sido concebida, codificada, testada e documentada na íntegra apenas pelos seus autores, que são os dois membros do grupo, **sem prejuízo de que os elementos do grupo podem reutilizar fragmentos de código das suas resoluções do projeto de Programação 1, caso o tenham realizado, que entenderem convenientes.**

Alunos detetados em situação de fraude, incluindo aquisição a terceiros ou plágio parcial ou total — tanto os plagiadores como os plagiados, com ou sem a intervenção de intermediários — em alguma componente de avaliação ficam liminarmente com essa 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 abordagens e algoritmos aplicáveis. Mas não da partilha de código. É 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 deteção de plágio que compara cada ficheiro na resolução de cada grupo com cada ficheiro das resoluções dos outros grupos. Como esperado para a sua funcionalidade, este software deteta tentativas de ludibriação, como, por exemplo, renomeação de variáveis, funções, constantes, etc, reposicionamento das funções no código, inserção de linhas em branco extra, etc, etc, etc. Evitem espertezas parvas, para evitarem desastres amargos.

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, enjoyTagusGroup00546.zip).

A **ÚNICA FORMA DE ENTREGA é a através do site da disciplina, em:**
<https://moodle.ciencias.ulisboa.pt/course/view.php?id=3943>

Qualquer entrega noutra forma não será considerada para avaliação.

Para ser avaliada, a vossa solução deve ser submetida até ao **PRAZO de sexta-feira, 24 de março de 2023, 23h00 (hora de Lisboa).**

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