Ciências ULisboa Faculdade de Ciências da Universidade de Lisboa

DEPARTAMENTO·DE·INFORMÁTICA

Programação 1 (LTI), 2023/2024

Projeto

(este enunciado tem 15 páginas)

birthPlan

0. Contexto

De acordo com registos estatísticos e comparações internacionais, num passado não muito distante, o Serviço Nacional de Saúde em Portugal alcançou um desempenho que o colocava entre os melhores do mundo. Tal desempenho surge hoje como uma imagem distante face aos hospitais com serviços de urgência encerrados, aos milhões de utentes sem médico de família, às filas de espera para atendimento nos centros de saúde que se começam a formar de madrugada, ao tempo de espera por uma consulta da especialidade, por vezes de anos, etc.

Para tal desempenho notável no passado, contribuíam de sobremaneira os serviços de excelência de apoio à maternidade e à natalidade, em tempos considerados como estando entre os melhores do modo. Atualmente, porém, as mães e os seus bebés enfrentam riscos acrescidos de saúde e de vida face ao encerramento, à intermitência e à imprevisibilidade dos serviços de maternidade prestados pelos diversos hospitais.

Nunca o montante disponível para o funcionamento do Serviço Nacional de Saúde foi tão elevado como atualmente, de acordo com a apresentação do orçamento de estado para 2024. Face a este cenário e face a este nível de financiamento, o próprio ministro da tutela converge com analistas em assinalar que um das principais causas destes problemas se deve a falta de organização.

Para tentar contribuir para a mitigação deste problema, um grupo de hospitais com serviços de maternidade tomou a iniciativa de recorrer aos alunos de Programação 1 (LTI) da Faculdade de Ciências de Lisboa para melhorar a sua organização e otimizar os recursos disponíveis. Estes alunos 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 `birthPlan`. É um software que apoia a equipa de enfermagem no encaminhamento de grávidas para os serviços de maternidade e que irá ser usado pela coligação de hospitais smartH, para gerir a atribuição de assistência no parto a médicos especialistas e respetivos hospitais e equipas.

Funcionalidade

O seu programa **recebe** uma **listagem dos médicos**. Esta listagem caracteriza, num dado momento, cada um dos médicos quanto aos aspetos relevantes para o atendimento de pedidos de assistência a partos. O seu programa recebe também um **calendário de assistências já agendadas** e ainda uma **listagem dos pedidos** de assistência que se encontram por atribuir a médicos desde a última calendarização.

O seu programa **entrega**, por um lado, o **calendário de assistências atualizado**. Por outro lado, entrega ainda a **listagem dos médicos atualizada**, após os pedidos de assistência terem sido analisados e eventualmente distribuídos por eles, conforme as disponibilidades dos mesmos e os requisitos dos pedidos feitos pelas enfermeiras.

Entrada

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

doctors14h00.txt

```
Organization:
SmartH
Hour:
14h00
Day:
08:12:2023
Doctors:
Manuel Frias, 2, 14h25, 85, 36h28
Carlos Sousa, 3, 12h10, 60, 28h34
...
```

schedule14h00.txt

```
Organization:
SmartH
Hour:
14h00
Day:
08:12:2023
Schedule:
14h00, Anabela Rocha, Carlos Sousa
14h25, Daniela Silva, Manuel Frias
...
```

Estes dois ficheiros de entrada contêm o resultado do planeamento realizado a uma dada hora. No exemplo acima, contêm o resultado do planeamento realizado às 14h00.

O outro ficheiro de entrada, exemplificado abaixo, contem o resultado do pedidos de assistência acumulados nos 30 minutos seguintes. No exemplo abaixo, contêm o resultado da recolha de pedidos até às 14h30.

requests14h30.txt

```
Organization:
SmartH
Hour:
14h30
Day:
08:12:2023
Mothers:
Zulmira Zacarias, 33, red, low
...
```

Saída

O programa produz dois ficheiros, um com a listagem dos médicos atualizada e outro com a calendarização das assistências atualizada.

Assume-se que estas duas atualizações são feitas em simultâneo e à hora do ficheiro de entrada com a recolha dos pedidos. Seguindo o exemplo acima, essa hora dos dois ficheiros de saída seria as 14h30.

O **ficheiro de saída com a listagem de médicos atualizada** tem uma estrutura interna similar ao ficheiro de entrada com a listagem dos médicos. 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 três campos de cada médico: dois campos com o número de horas acumulado, um desde a última

folga semanal, o outro desde o início do dia; e um outro campo com a data e hora de fim da mais tardia das suas assistências que já lhe estão atribuídas.

O **ficheiro de saída com a calendarização dos pedidos de assistências 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 assistência atribuídos a médicos por via da atualização foram acrescentados e os partos entretanto já realizados até ao tempo da atualização foram removidos.

Exemplos completos, com respetivos ficheiros de entrada e de saída, encontram-se nos test sets distribuídos com o presente enunciado.

Mais sobre especificação geral

- As diferentes listagens (médicos, calendarização, pedidos) são guardadas em outros tantos ficheiros .txt.

- Cada listagem começa com um cabeçalho que contém a indicação da organização, da hora da atividade, do dia da atividade, e do âmbito do ficheiro (`Doctors`, `Schedule` ou `Mothers`) como neste exemplo:

```
Organization:
SmartH
Hour:
14h00
Day:
08:12:2023
Schedule:
```

- 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:

  `doctors19h30.txt`

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

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

  Cada médico é caracterizado por: nome (e.g. Manuel Frias); a categoria profissional (de menor para maior experiência: `1`, `2`, ou `3`); a hora planeada a que terminará o último parto que irá dar assistência (e.g. `14h25`) (n.b.: para atribuição de nova assistência, contará a mais tardia de entre esta hora e a hora do cabeçalho); os minutos acumulados desde o início do dia, relativos aos partos que lhes estão atribuídos e que está a realizar ou ainda vai realizar

(e.g. `85`); as horas e os minutos acumulados desde o último descanso semanal, relativos aos partos que lhes estão atribuídos e que está a realizar ou ainda vai realizar (e.g. `36h28`); como ilustrado no seguinte exemplo:

```
Manuel Frias, 3, 14h25, 85, 36h28
```

Para cada médico, os tempos acumulados, do trabalho que está a realizar e que ainda vai realizar, aumentarão no momento em que lhe for atribuída uma assistência.

Se por efeito de uma atribuição, o tempo diário acumulado ultrapassar pela primeira vez os 240 minutos (4 horas), há que acrescentar 1 hora de pausa para se calcular a hora a que esse médico voltará a estar disponível, depois da pausa diária.

Se por efeito de uma atribuição, o tempo semanal acumulado ultrapassar o limiar de 40 horas, a linha correspondente a esse médico deve ficar como no exemplo seguinte:

```
Manuel Frias, 3, weekly leave, 160, 40h15
```

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 4h às 20h em todos os dias do ano; os serviços não param aos feriados ou fins de semana; todos os meses do ano de todos os anos têm 30 dias; apenas os períodos de pausa diários são levados em consideração; os períodos de folga semanal dos médicos e o seu impacto na calendarização são tratados por outra aplicação.

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

  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 médicos e da calendarização.

  Cada pedido é caracterizado pelo nome da mãe (e.g. `Anabela Rocha`); pela sua idade, em anos (e.g. 28); pela cor da pulseira (de menor para maior premência: `green`, `yellow` e `red`); pelo nível de risco do parto devido a comorbilidades ou outros fatores relevantes (de menor para maior risco: `low`, `medium` e `high`); como ilustrado no seguinte exemplo:

  ```
  Anabela Rocha, 28, yellow, high
  ```

  Cada assistência é planeada para durar 20 minutos.

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

  Cada pedido é caracterizado pela hora a que fica planeada a assistência ocorrer (e.g. `14h20`); pelo nome da mãe (e.g. `Anabela Rocha`); e pelo nome do médico assistente (e.g. Carlos Sousa); como ilustrado no seguinte exemplo:

  ```
  14h20, Anabela Rocha, Carlos Sousa
  ```

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

  Esse ficheiro de saída inclui as assistências já calendarizadas no ficheiro de entrada e que têm início na hora desse ficheiro de saída ou após essa hora. Todas as novas assistências calendarizadas nesse ficheiro de saída não podem ocorrer antes da hora desse ficheiro.

  Os pedidos de assistência devem ser atribuídos aos médicos que estejam disponíveis mais cedo e de acordo com a maior prioridade a dar às mães em circunstâncias mais prioritárias, como indicado em mais pormenor a seguir.

  A **mãe** para quem é primeiro procurada uma assistência será aquela que apresenta o risco mais elevado. Em caso de empate, é a mãe que tem a pulseira com maior premência. Em caso de empate ainda, é a mãe com maior idade. Em caso de empate ainda, é aquela cujo nome é primeiro de acordo com a ordem lexicográfica.

  O **médico** a quem é atribuído uma assistência será aquele que primeiro está disponível para realizar essa assistência, tendo em consideração que apenas médicos de categoria igual ou superior a 2 podem assistir em partos de risco elevado. Em caso de empate, é o médico que é mais categorizado. Em caso de empate ainda, é o médico a quem falta mais tempo até chegar à sua pausa. Em caso de empate ainda, é aquele que está a mais tempo do seu descanso semanal. Em caso de empate ainda, é aquele cujo nome é primeiro de acordo com a ordem lexicográfica.

  Essa atribuição é feita até se atingir o volume máximo de horas acumulado que esse médico pode trabalhar sem folga semanal, ou ultrapassar esse limite apenas para finalizar um último serviço que começou antes desse limite ter sido atingido. Caso contrário, essa assistência terá de ser atribuída a outro médico. Se não houver nenhum outro médico em condições de a realizar, essa assistência passa para outra rede de hospitais.

Essa atribuição é feita desde que que a realização dessa assistência não ultrapasse as 20h do dia respetivo. Se ultrapassar, essa assistência passa para outra rede de hospitais.

Caso não haja nenhum médico que satisfaça as condições do pedido, essa assistência passa para outra rede de hospitais.

Uma assistência que passa para outra rede de hospitais fica caracterizada como no seguinte exemplo:

```
HHhMM, Anabela Rocha, redirected to other network
```

HHhMM será a hora da atualização da calendarização (e.g. 14h00). Caso exista mais de uma deste género, com uma mesma HHhMM, tais linhas ficam ordenadas de acordo com as regras de prioridade no atendimento das grávidas.

- No ficheiro de saída com a calendarização das assistências atribuídas, estas são ordenados pela **hora 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 das mães.

- Assume-se que a **atualização** da listagem dos **médicos** e da **calendarização** dos partos 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 por contrato tem de ser a abordagem seguida.

Estrutura da aplicação

A aplicação `birthPlan` é composta pelo programa `refresh.py` e pelos seguintes módulos a que este recorre:

```
constants.py
dateTime.py
infoFromFiles.py
planning.py
infoToFiles.py
```

Os nomes destes módulos são auto-explicativos: o módulo `constants` define as constantes; `dateTime` contém as funções para lidar com formatos e operações com datas e tempos; `infoFromFiles` e `infoToFiles` contém

funções, respetivamente, para ler de ficheiros para estruturas de dados, e para escrever de estruturas de dados para ficheiros; e `planning` contém funções para realizar a calendarização e atualização das estruturas de dados.

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 `refresh.py` por sua vez contem uma função, com nome `plan`, cuja chamada assegura o funcionamento da aplicação.

Exceção
Deve ser lançada a exceção:

```
File head error: scope 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 âmbito (`doctors`, `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 refresh.py inputFile1 inputFile2 inputFile3
```

ATENÇÃO:
inputFile1 é um ficheiro com a listagem dos **médicos**.
inputFile2 é um ficheiro com a **calendarização das assistências**.
inputFile3 é um ficheiro com a listagem dos **pedidos de assistência**.

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 médicos, o nome de `doctorsXXhYY.txt`, 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 varíavel `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 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
- esqueletos dos módulos
- 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
Uma pasta com o ficheiro com:

- o relatório de implementação

e com os ficheiros de código desenvolvidos, incluindo os seguinte seis ficheiros (e outros desenvolvidos pelo aluno se for o caso):

```
constants.py
dateTime.py
infoFromFiles.py
planning.py
infoToFiles.py
refresh.py
```

A pasta deve ter o nome `birthPlanGroupN`, 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 1024, a pasta deve ter o nome `birthPlan1024`. A pasta tem de ser submetida **zipada**, com o nome `birthPlan1024.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:

```
#2023-2024 Programação 1
#Grupo 1024
#63224 Albertina Anacleta
#65552 Gervásio Gerónimo
```

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 de funções extra implementadas (se aplicável) e do 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 do programa 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 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 das funções completas, comentários q.b.), 10%
E. Legibilidade (nomeação de variáveis e funções perspícua, arrumação e formatação do código, espaçamento apropriado e consistente, sem *scroll* horizontal para ser lido, etc), 5%
F. Relatório de implementação, 5%

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

Classificação da componente de avaliação "Projeto"
O plano de avaliação nesta UC foi apresentado no início do semestre e encontra-se disponível desde então na respetiva página moodle.

Sobre a componente de avaliação "Projeto", relembra-se que engloba o **programa** (que deve procurar resolver o problema apresentado no presente enunciado) e respetivo **teste de aferição**.

Sobre o teste de aferição, relembra-se que se trata de prova escrita e individual, presencial e sem consulta, sobre tópicos da resolução do problema pelo programa. Será realizado nas datas dos exames juntamente com estes, uma única vez por cada aluno, não havendo lugar a melhoria.

Discrepância entre as classificações de cada membro de um grupo no teste de aferição será refletida em diferente classificação de cada membro no elemento de avaliação "Projeto" de acordo com procedimento na página da UC.

Discrepância entre a classificação no programa e as classificações dos seus autores no teste de aferição, poderá levar à realização de prova oral para averiguação da autoria e/ou da classificação a atribuir no elemento de avaliação "Projeto", de acordo com apreciação e decisão dos docentes, em data determinada por estes. Na há lugar a provas orais solicitadas pelos alunos para esta ou outras finalidades.

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.

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.

Pode e deve haver entreajuda 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, de funções, de constantes, etc, reposicionamento das funções no código, inserção de linhas em branco extra, etc, etc, etc.

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, `birthPlanGroup546.zip`).

A **ÚNICA FORMA DE ENTREGA é através do site da disciplina, em**:
`https://moodle.ciencias.ulisboa.pt/course/view.php?id=4742`

Qualquer entrega noutra forma não será considerada para avaliação.

Para ser avaliada, a vossa solução deve ser submetida até ao **PRAZO de sexta-feira, 15 de dezembro de 2023, 23h00 (hora de Lisboa)**.

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