UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
Departamento de Engenharia Informática

Programação Orientada aos Objetos

Projeto 2017/18
Serão Convívio de POO

Data de Entrega: 17 de Dezembro

Descrição do Problema

Os alunos de Programação Orientada aos Objetos (POO) pretendem organizar um serão de convívio com a comunidade do DEI e querem desenvolver uma aplicação de suporte ao planeamento do evento. A noite de convívio contempla a visita a vários locais em Coimbra incluindo parques, exposições e bares. A aplicação deve permitir realizar as seguintes operações:

- Inscrição no convívio
- Seleção dos locais a visitar
- Seriação dos locais a visitar
- Visualização de "guest lists"
- Contabilização das receitas dos locais visitados

Todos os locais são caracterizados pelas suas coordenadas GPS. Os parques podem ser jardins (caracterizados pela área) ou áreas desportivas (caracterizadas pelo(s) desporto(s) que se pode(m) praticar). As exposições são caracterizadas pela forma artística ("Pintura", "Escultura", ou outra) e pelo custo do ingresso. Os bares são caracterizados pela lotação e valor do consumo mínimo.

Os elementos da comunidade do DEI podem ser professores, funcionários ou estudantes. Cada pessoa tem um perfil que pode ser "Desportivo", "Cultural", "Boémio" ou "Poupadinho". As diferenças principais entre as pessoas são as seguintes:

- Os professores podem ser "Auxiliar", "Associado" ou "Catedrático"
- Os funcionários podem ser a "Tempo parcial" ou "Tempo integral"
- Os alunos pertencem a um curso.

A aplicação deve permitir realizar as seguintes operações:

1. Inscrição no convívio
   - A inscrição está restrita aos membros da comunidade do DEI
   - Cada pessoa deve inserir uma palavra chave para controlo de acesso

2. Apresentação dos locais a visitar
   - Após a inscrição, a aplicação deve mostrar a lista dos locais que podem ser visitados

- A aplicação dever permitir visualizar o número atual de inscritos em cada local e a lotação (quando se aplica)

3. Seleção dos locais a visitar
   - Cada pessoa inscrita deve selecionar até 5 locais que pretende visitar durante o serão de convívio

4. Seriação dos locais a visitar
   - A aplicação deve permitir apresentar uma lista ordenada de locais a visitar por ordem decrescente do número de inscrições

5. Visualização da “guest lists”
   - A aplicação deve permitir visualizar uma “guest list” para cada bar
   - O número máximo de convidados em cada “guest list” corresponderá a uma percentagem da sua lotação, a definir na configuração da aplicação
   - A introdução das pessoas nas “guest lists” é realizada por ordem de inscrição. No entanto, as pessoas com perfil “Boémio” têm prioridade no acesso às “guest lists” podendo substituir utilizadores com outro perfil
   - Cada pessoa na “guest list” deve ser apresentada com o seguinte formato:
     - Aluno: nome, perfil e curso
     - Professor e Funcionário: nome e perfil

6. Contabilização das receitas dos locais visitados
   - A aplicação deve permitir apresentar o valor mínimo de receita prevista para o evento tendo em consideração as inscrições existentes e os locais a frequentar
   - Deve considerar que os alunos têm um desconto de 10% no preço da entrada nas exposições

A interação com o utilizador deverá ser realizada através de uma **interface gráfica**. A aplicação deve ser disponibilizada com **ficheiros de texto** contendo dados relativos à comunidade e locais a visitar (pelo menos 10 elementos de cada categoria). Estes ficheiros devem apenas ser lidos quando ainda não houve inscrições. Após a realização da primeira inscrição, todos os dados devem ser guardados em **ficheiros de objetos** e carregados sempre que a aplicação for iniciada.

Implementação

A aplicação deve ser implementada na linguagem Java e deverá ter em conta os seguintes aspetos:

1. Cada classe deve gerir internamente os seus dados, pelo que deverá cuidar da proteção das suas variáveis e métodos

2. Cada objeto deverá ser responsável por uma tarefa ou objetivo específico, não lhe devendo ser atribuídas funções indevidas

3. Utilize a *keyword* **static** apenas quando tal se justifique e não para contornar erros do compilador

Elabore um diagrama com as suas classes (em UML) antes de iniciar a implementação, para prever a estrutura do projeto.

Tenha ainda em conta os seguintes pontos que serão importantes na avaliação:

1. Comentar as classes, métodos e variáveis públicas segundo o formato Javadoc. Isto permitir-lhe-á gerar automaticamente uma estrutura de ficheiros HTML, descritivos do seu código, que deve incluir no seu relatório

2. Comentar o restante código sempre que a leitura dos algoritmos não seja óbvia

3. Tal como sugerido acima, evitar o uso abusivo de **static** e de variáveis e métodos **public**

4. Na escolha de nomes para variáveis, classes e métodos, seguir as convenções adotadas na linguagem **Java**

5. Na organização das classes deverá ser evitada a redundância dos dados

Prazos de entrega

A entrega do trabalho compreende duas metas distintas:

Meta 1 (1 valor) – Análise do projeto e diagrama de classes (entrega até **5 de Novembro**);

Meta 2 (4 valores) – Versão final (entrega até **17 de Dezembro**).

MUITO IMPORTANTE:

Os trabalhos serão comparados (tanto entre os trabalhos da disciplina como com código disponível na Internet), no sentido de detetar eventuais fraudes por cópia. Nos casos em que se verifique que houve cópia de trabalho total ou parcial, os grupos envolvidos terão os projetos anulados, reprovando à disciplina. Serão aplicadas as regras da Universidade de Coimbra relativamente a plágio.

Material a entregar

Cada grupo deve entregar obrigatoriamente:

Meta 1: Diagrama de classes em UML

1. Upload em pdf do diagrama de classes no InforEstudante

2. Entrega de uma cópia impressa do diagrama no momento da defesa na aula teórico-prática

Meta 2: Aplicação

1. Upload de zipFile no InforEstudante com:
    - Todas as classes .java
    - Executável
    - Ficheiros de dados para teste
    - JavaDoc
    - Relatório (em formato pdf)

2. Relatório em papel, entregue no cacifo do docente da turma teórico-prática a que os alunos pertencem, descrevendo o programa do ponto de vista técnico e que deve incluir:
    - Estrutura geral do programa
    - Diagramas de classes inicial e final
    - Descrição das principais estruturas de dados e de ficheiros usados
    - Breve explicação de como o programa se executa

Avaliação do trabalho

Para a avaliação do trabalho contam fatores de dois tipos:

- Caixa preta (tal como é percecionado pelo utilizador):
    - Conjunto de funcionalidades implementadas;
    - Robustez do programa;
    - Qualidade da interface.
- Caixa branca (a forma como está construído):
    - Qualidade das soluções técnicas encontradas para os problemas em causa;
    - Estruturação do código;
    - Qualidade dos comentários.

A avaliação em cada uma das metas é feita individualmente, independentemente dos grupos serem constituídos por 2 alunos.

Nota: Não se aceitam trabalhos que apresentem erros de compilação no momento da defesa e que não estejam corretamente estruturados do ponto de vista da Programação Orientada aos Objetos.

Composição dos grupos

O trabalho deve ser realizado em grupos de 2 elementosda mesma turma

teórico-prática. A defesa dos trabalhos é feita individualmente, bem como a respetiva avaliação.

Defesa final do trabalho

O trabalho deve ser defendido através de uma discussão presencial e individual. Para isso, cada grupo deve inscrever-se num horário que esteja disponível para essa defesa no InforEstudante.