Trabalhos Práticos

Algoritmia e Programação

Engenharia Informática - 1º ano 1º semestre
Ano Lectivo 2005/2006

---

1. Objectivos
2. Calendarização
3. Normas
3.1 Relatório de Progresso
3.2 Relatório Final
3.3 Avaliação
4. Propostas

---

1. Objectivos

Análise, projecto e desenvolvimento de uma pequena aplicação que envolva alguns dos conceitos abordados na disciplina, nomeadamente, estruturas e vectores.

O trabalho prático deve ser realizado em grupo (máximo 2 elementos), extra aulas. No final o trabalho será acompanhado de um relatório e será apresentado e demonstrado individualmente. Este trabalho tem um peso de 60% na Nota de Frequência (NFREQ).

Notas:

- Para os alunos com dispensa de avaliação contínua é obrigatória a realização do trabalho prático.
- A Nota de Frequência só é válida no semestre em que foi obtida.

2. Calendarização

Lançamento das propostas de trabalhos: até 4 de Novembro de 2005
Constituição dos grupos e temas escolhidos: até 11 de Novembro de 2005
Entrega do relatório de progresso : até 2 de Dezembro de 2005
Entrega do trabalho: até 2 de Janeiro de 2006
Apresentação e discussão: de 2 a 6 de Janeiro de 2006

A identificação dos grupos e temas escolhidos, assim como o relatório de progresso e o trabalho deve ser entregue ao professor das aulas práticas.

3. Normas

O trabalho pode ser realizado individualmente ou em grupos de dois alunos, no entanto, a apresentação e discussão será sempre realizada individualmente.

1. A data final de ENTREGA do trabalho é 2 de Janeiro de 2006. No entanto os grupos terão de cumprir as seguintes fases intermédias:

- Semana de 7.11.05 a 11.11.05: **Identificação do grupo e descrição informal** (mas escrita) do trabalho. Um grupo que não cumpra este prazo terá uma penalização de 1 valor na nota final do trabalho.
- Semana de 28 de Novembro a 2 de Dezembro: **Relatório de Progresso**, com uma descrição das estruturas de dados que pretendem utilizar e uma descrição geral da organização do programa. Um grupo que não cumpra este prazo terá uma penalização de 2 valores na nota final do trabalho.
- Até 2 de Janeiro de 2006: **Entrega do trabalho**. Um grupo que não cumpra este prazo terá uma penalização de 3 valores na nota final do trabalho. **NÃO SERÃO ACEITES** trabalhos depois de 4 de Janeiro de 2006.

Independentemente destes prazos, os grupos deverão ser capazes de, quando o professor o solicitar, reportar o estado de desenvolvimento do trabalho.

A entrega do trabalho consta de um relatório (ver estrutura no ponto seguinte), código fonte do programa e executável.

2. A apresentação e discussão do trabalho decorrerá na semana de 2 a 6 de Janeiro, em dia e hora marcar por cada professor das práticas.

- No dia da apresentação, TODOS os elementos do grupo deverão estar presentes. Os elementos ausentes não terão classificação.

3. Cada grupo é responsável por gerir o seu processo de desenvolvimento. Dificuldades e problemas deverão ser comunicadas atempadamente ao professor das aulas práticas.

3.1 Relatório de Progresso

Dever ser realizado um relatório de progresso, onde deverá constar obrigatoriamente:

- a identificação do trabalho,
- a identificação completa dos alunos que o realizaram,
- a definição das estruturas de dados utilizadas,
- uma explicação da estrutura geral do programa

e qualquer outra informação adicional que o aluno julgue conveniente.

3.2 Relatório Final

Do relatório final deverão constar a identificação do trabalho e a identificação completa dos alunos que o realizaram. Este relatório deverá seguir as seguintes recomendações de estrutura:

1. Introdução – Identificação e descrição do trabalho e objectivos propostos.

2. Estruturas de Dados – definição das estruturas de dados utilizadas no programa.

3. Algoritmos - Apresentar o algoritmo de funcionamento global do programa. Para as funcionalidades fundamentais do programa apresentar o(s) algoritmo(s) envolvidos. Os algoritmos devem ser descritos em Português Estruturado ou Fluxogramas.

4. Manual do Utilizador – Descrição das funcionalidades do programa apresentadas ao utilizador.

5. Conclusões - Apresentação das dificuldades que foram sentidas na execução do trabalho. Limitações do programa desenvolvido e funcionalidades que poderiam/deveriam ser modificadas/melhoradas.

6. Anexos

3.3 Avaliação

Na avaliação do trabalho serão considerados

- a concretização dos objectivos propostos,
- e definição das estruturas de dados,
- a estruturação do programa,
- a robustez,
- o interface.

4. Propostas

Tema 1 – Gestão de Colecções de Selos

Tema 2 – Gestão e Controlo dum Parque de Estacionamento

Tema 3 – Gestão de Classificações

Tema 4 – Gestão do Processo de Calendarização de Exames

Tema 5 - Gestão de uma Escola de Natação

Tema 6 – Gestão de uma Empresa de Telecomunicações

Tema 7 – Sistema de Gestão de Stocks

Tema 1 – Gestão de Colecções de Selos

Enquadramento: Os coleccionadores de selos de forma a aumentarem as suas colecções realizam trocas de selos (repetidos) com outros coleccionadores. Tradicionalmente estas trocas são feitas com correspondentes no estrangeiro. No entanto, a gestão destas trocas exige grande organização e memória, designadamente, para saber em cada momento quais os selos que já foram enviados e para quem, de forma a evitar repetições nos envios.

Objectivo: Desenvolver um programa em linguagem C^{++} que permita efectuar a gestão de uma colecção particular de selos.

Tarefas: Conceber as estruturas de dados e algoritmos necessárias à implementação da referida aplicação em linguagem C^{++}.

Entre outros conceitos, "**selo**" e "**correspondente**" são de particular importância na aplicação a desenvolver, e devem incluir as seguintes propriedades:

Estrutura: Selo	**Exemplo**
{NumeroSerie, Nome, Serie, Cor, Ano, Largura, Altura, Valor_Facial, QtdStock}	{43, "Atletismo", "Jogos Olímpicos 2000", "Amarelo", 1999, 12, 18, 2.45, 3}

Estrutura: Correspondente	**Exemplo**
{NumeroSerie, Nome, Morada, DataInicio,UltimoEnvio}	{21, "John Bull", {Kensington Street, 45; London-41388; England}, {02/03/1999}, {14/01/2003}}

Esta aplicação deve permitir:

- Adicionar selos à colecção;
- Alterar a quantidade existente de um selo;
- Adicionar e alterar um correspondente;
- Proceder a um envio, isto é, armazenar os dados relativos ao envio de uma lista de selos para um determinado correspondente;
- Propor um envio, isto é, gerar uma lista de selos para enviar para um determinar correspondente que não tenham sido enviados anteriormente, ordenados pela quantidade existente em stock;

Tema 2 – Gestão e Controlo dum Parque de Estacionamento

A direcção da escola "*Mentes Brilhantes*" decidiu propor aos alunos de APRO a realização dum programa para controlar os acessos ao seu parque de estacionamento.

Existem cancelas à entrada e à saída do parque, que tem 20 lugares de estacionamento.
Para entrar ou sair é necessário inserir um cartão (contendo o código do utilizador) no respectivo controlador de cancela. O controlador de entrada responde com:

- *"Avance para o lugar X"*, emitindo um talão com o número do lugar atribuído, que seria colocado no *tablier* do automóvel
- ou "Sem acesso", seguido do motivo: *código inválido, lotação completa.*

Existem 4 categorias de utentes do parque, cada uma delas com diferentes privilégios de acesso:

- *Direcção*
- *Professores e funcionários*
- *Alunos*
- *Visitantes*

Existem 3 tipos de acesso:

- *Lugar cativo*
- *Prioritário*
- *Normal*

Os utentes com acesso cativo têm sempre lugar.
Os utentes prioritários e normais têm acesso de acordo com as seguintes regras:

- Se a percentagem de lugares ocupados, calculada com base no total de lugares de estacionamento exceptuando os cativos, for inferior a 50% então ambos têm acesso
- Se a percentagem de lugares ocupados for superior ou igual a 50% então só os utentes prioritários têm acesso
- Quando a lotação esgota ninguém entra (excepto os que têm lugar cativo, bem entendido).

O limiar de 50% poder ser **alterado** ao longo do programa, variando entre 0-100.

A tabela seguinte apresenta os privilégios de acesso correspondentes a cada categoria de utente:

	Direcção	*Profs eFuncs*	*Alunos*	*Visitantes*
Lugar cativo	X			
Prioritário	X	X		
Normal		X	X	X

O número máximo de visitantes simultâneos é **configurável** pelo utilizador.

Os códigos de utente devem ser atribuídos sequencialmente a partir de zero.
Os códigos de utentes que foram eliminados da base de dados não podem ser reutilizados.

Funcionalidades mínimas requeridas

A base de dados de utentes deverá conter, pelo menos:

- Código
- Nome
- Categoria
- Número do lugar de estacionamento (**-1** se não estacionado)

- Multas
  - Data
  - Hora
  - Montante
  - Pago (sim/não)

Deverão existir 3 modos de funcionamento:

- *Modo administrativo:* serve para alterar a configuração do sistema.
  Por exemplo:
  - **Configurar** o limiar de acesso, o número de visitas simultâneas, o valor das multas, os lugares cativos, etc.
  - Inserir, listar, procurar, alterar ou eliminar dados dos utentes
  - Tratamento de multas
    - Pagamento
    - Identificação do utente mais faltoso
    - Valor total cobrado em multas num dado mês
  - Saber o lugar atribuído a um utente estacionado
- *Modo portão:* **simula** a entrada e saída de utentes, com visualização em modo texto do estado do parque (por exemplo: lugares ocupados/livres, percentagem de ocupação, etc.).
  Duas operações possíveis:
  - *Tentativa de entrada:* o utilizador indica o código do utente que pretende entrar. O programa deverá responder com "avance para o lugar X", ou "sem acesso" e qual a razão.
  - *Saída:* será fornecido o código do utente. Se esse utente não constar na base de dados como incorrectamente estacionado será aberta a cancela (em sentido figurado). Caso contrário só poderá sair após pagamento da respectiva multa (10€ (**configurável**)).
- *Modo inspecção:* permite identificar manualmente os utentes mal estacionados (indicando os respectivos códigos e a data/hora da ocorrência)

Deve-se ter em conta que as alterações administrativas podem causar problemas durante a operação do parque. Por exemplo, como eliminar um utente se ele ainda estiver estacionado? Serão valorizadas as soluções mais inteligentes.

Tema 3 – Gestão de Classificações

Pretende-se gerir a informação relativa às classificações dos alunos do Curso de Eng. Informática. Para tal é necessário dispor de informação relativa aos alunos, ao plano de estudos do curso e às classificações obtidas pelos alunos nas várias disciplinas do curso.
Para caracterizar o aluno é necessário dispor do número, nome, data de nascimento, morada e telefone.
O plano de estudos do curso deve conter informação sobre as disciplinas que integram cada semestre de cada ano e os respectivos créditos.
Para gerir as classificações obtidas pelos alunos será preciso armazenar informação sobre a nota final, a nota de frequência e a nota de exame (tendo em atenção que pode ser época normal ou recurso), a disciplina a que diz respeito e o respectivo ano lectivo.

Elabore um programa que permita:

1. Efectuar a manutenção dos alunos, disciplinas do curso e notas (inserção, alteração e eliminação)

2. Apresentar as seguintes listagens:
    - As classificações finais obtidas em todas as disciplinas de um determinado ano do curso por um dado aluno.
    - As disciplinas e o respectivo ano cuja média, num dado ano lectivo, foi superior a um determinado valor
    - A percentagem de alunos que obtiveram classificação final positiva numa determinada disciplina num ano lectivo
    - Os nomes dos alunos que cuja média global das disciplinas feitas (com nota final positiva) é superior a um qualquer valor lido do teclado (somatório das notas das disciplinas multiplicadas pelos respectivos créditos e dividir pelo somatório dos créditos dessas disciplinas)
    - As notas de todos os alunos, identificados pelo seu nome, numa determinada disciplina, num dado ano lectivo por ordem decrescente

Tema 4 – Gestão do Processo de Calendarização de Exames.

O Departamento de Engenharia Informática do ISEP deseja informatizar o seu processo de calendarização de exames.
Para tal deve:

- Criar um vector de strings com a Sigla;
- Criar um vector de strings com o nome das disciplinas;

Cada disciplina tem a indicação do respectivo ano.
Cada Docente tem a sua Sigla e a indicação da sua carga horária.
Deve também ser criada uma matriz de índices que permita relacionar os Docentes com as disciplinas, com a indicação se é, ou não regente.

Pretende-se uma aplicação que por cada época de exames permita:

- Criar um calendário de exames entre duas datas, tendo em atenção que no mesmo dia não podem existir exames de disciplinas do mesmo ano e tendo em conta que não podem decorrer em simultâneo exames de disciplinas leccionadas pelo mesmo Docente;
- A partir do calendário de exames faça uma função que permita fazer a distribuição de vigilâncias tendo em conta que por cada disciplina deve existir um número de seis Docentes.
  Nota: Os Docentes da disciplina estão automaticamente convocados.
  Existe a seguinte regra de distribuição de vigilâncias:

De 1 a 3 horas de serviço -> 0 a 2 vigilâncias
De 4 a 6 horas de serviço -> 3 a 4 vigilâncias
De 7 a 10 horas de serviço -> 4 a 5 vigilâncias
Mais de 10 horas 6 vigilâncias

Pretende-se também que a aplicação permita listar as vigilâncias por Docente ou por disciplina.
Para além das tradicionais opções de adição, alteração e remoção deve também ser dada a opção ao regente da disciplina de dispensar um Docente da vigilância.

Tema 5 - Gestão de uma Escola de Natação

A escola de natação "NADA SENÂO AFOGASTE" pretende uma aplicação informática para gerir os alunos da escola. Existem quatro Técnicas de natação:

- Técnica de natação Crol;
- Técnica de natação Costas;
- Técnica de natação Bruços;
- Técnica de natação Mariposa.

Requisitos:

- As aulas têm uma duração 45m de aula e um intervalo de 15m.
- A escola está aberta de 2ª a 6ª feira das 8h até ás 13h e das 15h às 20h.
- A piscina utilizada para ministrar as aulas tem dois tanques, portanto pode haver duas aulas em simultâneo. Os professores são pagos por aula dada.
- Os alunos estão organizados em turmas. Cada turma tem um professor e um horário. A cada turma só é ensinada uma única técnica. Existe um número máximo de alunos por turma.
- As aulas têm um custo mensal (os alunos com de 65 anos tem um desconto de 50%). Caso o aluno não paga dois meses é-lhe interdito o acesso à piscina.
- Os alunos podem mudar de turma.
- O numero de dias de aula varia de mês para mês.

A aplicação deve efectuar as seguintes operações:

- Professores:
  - Inserir, alterar, eliminar informação;
  - Mostrar horário, as turmas, número de alunos…
- Alunos:
  - Inserir (os alunos tem que pertencer a uma turma), alterar e eliminar informação
  - Registar os pagamentos e alertar para as situações de incumprimento.
- Escola:
  - Mostrar o número de alunos (total, por professor, por turma e com mais de 65 anos).
  - Calcular despesas e receitas (total e por professor) a manutenção da piscina tem um custo fixo mensal.

Tema 6 – Gestão de uma Empresa de Telecomunicações

A empresa de telecomunicações "ESTOU SIM É PRA MIM" pretende uma aplicação para gerir as facturas dos clientes. Esta empresa é uma empresa de telecomunicações móveis. O número de telemóvel desta empresa tem o seguinte formato: 90xxxxxxx. Tem nove dígitos em que os primeiros dois são o '9' e o '0'.
A informação registada quando um cliente faz uma chamada é a apresentada na tabela seguinte.

DATA	HORA	DESTINO			DESCRITIVO	DUR CHAM	TIPO TARIFA	CUSTO
		INDIC. PAÍS	INDIC. NACIONAL	NÚMERO				
20040409	17:54:00	351	91	9004080	Vodafone	00:01:00	Fim Semana	1.00 €
20040409	18:01:00	351	91	9004080	Vodafone	00:01:00	Fim Semana	1.00 €
20040409	19:15:00	351	91	9004080	Vodafone	00:02:00	Fim Semana	1.00 €
20040409	19:58:00	351	91	9004080	Vodafone	00:01:00	Fim Semana	1.00 €
20040409	12:02:00	351	22	6174266	Porto	00:01:00	Fim Semana	1.00 €
20040409	17:44:00	351	22	6174266	Porto	00:02:00	Fim Semana	10.00 €
20040409	18:02:00	351	22	6174266	Porto	00:02:00	Fim Semana	0.00 €
20040409	19:12:00	351	22	6174266	Porto	00:01:00	Fim Semana	1.00 €

A empresa tem três tipos de tarifários (planos): Fim-de-semana, noite (entre as 21.00 e as 09.00) e normal, obviamente com preços diferentes.
Para a empresa sempre que um cliente faz uma chamada o custo é fixo independentemente do plano do cliente.
A aplicação deve:

- Clientes:
  - Inserir, alterar, eliminar informação;
  - Mostrar as chamadas efectuadas
  - Listar por ordem decrescente os números mais marcados
  - Listar por ordem decrescente os números com mais tempo de conversação marcados
- Chamadas:
  - Registar, eliminar e alterar a informação.
  - Estatística:
    - Percentagem de chamadas realizadas para clientes da empresa
    - Percentagem de chamadas realizadas para clientes de outras empresas (descriminado por empresa).
    - Plano com mais chamadas realizadas
- Empresa
  - Despesas e receitas.
  - Melhor cliente
  - Listar os clientes por ordem decrescente de chamadas efectuadas.

Tema 7 – Sistema de Gestão de Stocks

Conceba e implemente em C++ um sistema para a Gestão de Stocks de Componentes de um complexo fabril, o sistema **SGS**, com os seguintes requisitos:

- Cada linha de produção fabrica produtos de diversos tipos, tendo cada um deles um código. Em cada momento, a fábrica tem várias linhas de produção activas, devendo cada uma destas fabricar um certo número de unidades de um dado produto. Esta informação varia ao longo do tempo.
- O número de unidades do produto já produzidas em cada linha deve ser registada para que, a cada momento, se possa estimar o que falta ainda produzir.
- Para cada produto fabricável é conhecida a sua composição, ou seja, a lista dos componentes (contendo o seu código e o número total) necessários para fabricar 1 unidade desse produto.
- Cada componente possui um código único, uma descrição, uma quantidade em stock e um nível mínimo de quantidade em stock (designado por nível de alarme).
- Cada produto possui um código de produto único, uma descrição e uma definição da sua estrutura de componentes.

A aplicação deve permitir efectuar as seguintes operações:

- Inserir, mostrar, alterar e eliminar componentes
- Inserir, mostrar, alterar e eliminar produtos
- Registar a entrada em stock de *N* unidades de um dado componente
- Registar a saída de stock de *N* unidades de um dado componente
- Abrir a linha de fabrico nº *L* para produzir *X* unidades do produto *P*
- Registar a produção de *X* unidades na linha de fabrico nº *L*
- Listar todos os produtos
- Pesquisar para um dado produto os seus componentes e respectivas quantidades
- Listar a percentagem de fabrico actual de cada uma das linhas de produção em actividade
- Listar todos os componentes em nível de alarme
- Listar todos os produtos que não podem ser fabricados por falta de componentes
- Listar todos os componentes e respectivas quantidades, necessários para fabricar *X* unidades de um dado produto
- Listar as quantidades de componentes a encomendar para, em função do stock existente, se poder fabricar *X* unidades de um dado produto