



Disciplina: Estrutura de Dados e Orientação a Objetos

Curso: Redes de Computadores

Professora: Dra. Alana Moraes

Lista de Exercícios

1. Uma agência bancária estatal está sofrendo com o mau atendimento aos clientes. Suas filas são gigantescas! O gerente decidiu contratá-lo para que você crie uma simulação do atendimento da agência e permita que ele faça experimentos para melhorar o atendimento. O gerente vai informar a quantidade de minutos que ele deseja executar a simulação. Na agência existem 5 caixas. A cada minuto chegam de 4 a 16 clientes. Cada caixa atende de 1 a 2 clientes por minuto. Sempre o cliente dá preferência para alguma caixa vazia ou para uma fila com menor número de clientes. Para cada cliente que entra na fila deve-se registrar o momento (tempo) que ele entrou na fila. Quando o cliente for atendido deve-se computar a diferença entre o tempo atual e o tempo de entrada na fila. Este dado vai servir para calcular o tempo médio dos clientes na fila. O programa deve informar, por minuto, a quantidade de clientes sendo atendidos e esperando na fila para cada caixa. No término da simulação deve ser informado o tempo médio dos clientes aguardando na fila.

2. Manoel percebeu que seu estacionamento com uma entrada era um fiasco. Ele resolveu vender o seu terreno e comprar um novo que possui uma entrada e uma saída no fundo do terreno. Quando chega um novo carro, este é estacionado no terreno de Manoel, um atrás do outro. Quando um carro precisa sair, os carros do terreno são retirados pela saída, dão uma volta na quadra e são colocados no final da fila pela entrada do estacionamento. Faça um sistema que inclua carros no estacionamento informando o número da placa e retire carros usando o identificador (placa). Depois de ter informado a placa, cada vez que é pressionada a tecla S deve ser mostrado o estado do estacionamento.

3. Um hospital de cardiologia precisa de um sistema para efetuar o cadastro de pacientes que necessitam de doação de coração. Para cada paciente que é incluído no sistema deve ser informado o nome, telefone e o grau de urgência para transplante. O grau de urgência é definido na seguinte escala: (5) Muito urgente; (4) Urgente; (3) Médio; (2) Pouco urgente; (1) Sem urgência. Sempre que é o hospital recebe um novo coração o sistema é consultado para obter o próximo paciente que deverá ser operado. O sistema informa o nome e o telefone do paciente. Também a qualquer momento é possível visualizar o tamanho da fila de espera. Observação: os dados não precisam ser persistidos em arquivos, podem ficar

armazenados somente na memória. Se desejar que o arquivo seja salvo use o procedimento de arquivos aleatórios.

4. Você está criando uma nova engine orientada a eventos. Podem ocorrer três tipos de eventos: (a) Entrada de Dados; (b) Lógica; (c) Processamento Gráfico. Os eventos do grupo (c) são os que possuem maior prioridade, seguidos do grupo (b) e (a). Ou seja, a engine sempre dará preferência para executar eventos de Processamento Gráfico. Cada grupo é formado por um conjunto de eventos, cada um com uma determinada prioridade. Veja tabela abaixo:

Prioridade	Proc. Gráfico	Lógica	Entrada de Dados
<i>Alta</i>	Colisão	Lógica do jogo	Teclado
<i>Média</i>	Atualização da Tela	Inteligência Artificial	Mouse
<i>Baixa</i>	Partículas	Simulação Física	Internet

Por exemplo, se forem gerados os eventos Colisão, Internet e Partículas, primeiramente seria executado o evento Colisão, seguido de Partículas e por último Internet. Faça um programa que para uma determinada quantidade de tempo a cada fatia de tempo sejam produzidos aleatoriamente de 1 a 20 eventos descritos na tabela acima. Conforme são gerados os eventos estes devem ser mostrados na tela e inseridos na estrutura que vai solucionar este problema (dica: resolva este problema com uma combinação de filas e pilhas com prioridades). Concluída esta etapa cada evento será extraído da estrutura para ser executado (isto vai ser simulado, não precisa ser implementado). Conforme vai sendo extraído exibe-se uma mensagem na tela que o determinado evento está sendo executado. A cada fatia de tempo todos os eventos são tratados deixando a estrutura vazia.

5. Inverta a ordem dos elementos na pilha S criada pelo usuário:

- usando pilhas adicionais;
- usando uma fila adicional;

6. Ponha os elementos da pilha S na ordem ascendente usando uma pilha adicional e algumas variáveis adicionais.

7. Transfira os elementos da pilha S1 para a pilha S2 de modo que os elementos de S2 estejam na mesma ordem que em S1:

- usando uma pilha adicional;
- não usando pilha adicional, mas somente algumas variáveis adicionais.

8. Escreva um programa que determine se dada cadeia de caracteres é um palíndromo ou não, isto é, se pode ser lida da mesma maneira para frente ou para trás. Em cada ponto você pode ler somente um caractere da cadeia dada; não use uma matriz para primeiro armazenar esta cadeia e então analisá-la (exceto, possivelmente, em uma implementação de pilha). Considere usar pilhas múltiplas.

9. Escreva um programa para converter um número de notação decimal em um expressado em um sistema de número cuja base (ou raiz) é um número entre 2 e 9. A conversão é realizada pela divisão repetitiva pela base à qual um número está sendo convertido e, então, tomando-se os restos da divisão na ordem inversa. Por exemplo, para converter para binário, o número 6 exige três de tais divisões: $6/2 = 3$ com resto 0, $3/2 = 1$ com resto 1 e, finalmente, $1/2 = 0$ com resto 1. Os restos 0, 1 e 1 são colocados na ordem inversa, de modo que o binário equivalente de 6 é igual a 110. Modifique seu programa de modo que possa realizar uma conversão quando a base for um número entre 11 e 27. Os sistemas de números com bases maiores do que 10 exigem mais símbolos. Em consequência, use letras maiúsculas. Por exemplo, um sistema hexadecimal exige 16 dígitos: 0, 1, ..., 9, A, B, C, D, E, F. Neste sistema, o número decimal 26 é igual a 1A na notação hexadecimal, pois $26/16 = 1$ resto 10 (isto é, A) e $1/16 = 0$ resto 1.