

	<p>Universidade Federal de Campina Grande Departamento de Sistemas e Computação Disciplina: Técnicas de Programação</p> <p>PROVA DO MÓDULO I – LABORATORIAL – TURMA: 01 SEMESTRE 2016.2 – DATA: 20/12/2016</p>	
<p>Matrícula</p>	<p>Nome</p>	<p>Nota</p>

1. Em um universo paralelo disparatado, Lion, rei dos Thundercats acaba de anunciar o projeto de dois novos modelos que revolucionarão a relação custos-benefícios da indústria de naves de Thundera, no tocante ao consumo de combustível *versus* a distância percorrida. Contudo, somente um modelo poderá ser lançado no mercado no final deste mês. A fim de que o novo modelo realmente cumpra com o que é prometido em sua propaganda, é necessária a simulação do dilema da fundação em um programa em C++ que utilize o conceito de classe. Considerar que: (i) dois protótipos serão testados, cujos tanques principais de combustível têm capacidades máximas de 480 e 137 litros; (ii) o protótipo 1 consome 99 l/h, enquanto o protótipo 2 consome 58 l/h; e (iii) o combustível utilizado pelo protótipo 1 custa 7/4 do valor daquele utilizado pelo protótipo 2.

Diante do exposto, construir uma classe Thunder nave e implementar suas funções-membros de modo a poder testá-la na seguinte situação: (i) abastecimento do protótipo 1 com 435 l de combustível e protótipo 2 com 99 l; (ii) vôo de 55 minutos do protótipo 1 e de 49 minutos do protótipo 2, sobre linhas paralelas de vôo, partindo da Toca dos Gatos.

1,4

Especificação de requisitos:

- Abastecimento com uma quantidade de combustível (em litros), a ser definida pelo piloto;
- Simulação de deslocamento do veículo durante um dado intervalo de tempo (em minutos), a ser definida pelo piloto;
- Registro em arquivo de todos os dados da simulação do piloto, acrescidos da informação de custo-benefício associada a cada veículo considerado e um parecer sobre qual veículo se afigura mais vantajoso para o comprador da nave;
- Utilização de funções-membros para a manipulação dos objetos; e
- Emprego de encapsulação de dados.

Generalizar a implementação para receber da entrada-padrão quaisquer pares de valores, contidos em faixas válidas de quantidades de combustível, intervalos de tempo de voo e relação de custos entre combustíveis.

2. Observou-se que um dos grandes açudes da região do Curimataú paraibano inunda uma plantação de verduras existente em uma das margens do açude, caso a altura

de sua lâmina d'água ultrapasse 150 pés e 10 polegadas. Em virtude de diversos fatores ambientais, a altura da lâmina d'água do açude não é temporalmente constante, mas é possível medi-la no início de cada mês. A localidade é inundada durante a estação chuvosa, que vai de Abril a Julho e de Novembro a Dezembro, quando se espera um média de 6 polegadas de chuva por mês. Por outro lado, durante a estação seca, que ocorre nos demais meses, espera-se uma média de 3 polegadas de chuva por mês.

A partir desta descrição, escrever um programa que, a partir do mês e da altura observada da lâmina d'água do rio (em pés), no início daquele mês, emita uma mensagem comunicando a possibilidade (ou não) de inundação da localidade supramencionada, empregando a pluviosidade média do mês. [Considerar que nenhuma outra variável causa impacto neste modelo computacional] **1,3**

Exemplo: Entrada: *Digite o numero do mes de interesse: 05*
Digite a altura observada no mes 05 (em pés): 148.95
Saída: *Possivel inundacao!*

3. Em menos de uma semana da abertura do **TruFe (Truste de Férias)** por um respeitável grupo de pequenos acionistas da turma 01 de Técnicas de Programação do período 2016.2 da UFCG, o ministro da finanças brasileiro solicitou à turma a construção de um programa C++ para simular uma classe *Adesao*, constituída dos seguintes atributos: (i) *nome*; (ii) *data de nascimento*; (iii) *endereço* (composto por *nome da rua*, *número da casa*, *bairro* e *CEP*); (iv) *número de telefone fixo*; (v) *número de telefone móvel*; (vi) *número do CPF*; (vii) *id*; (viii) *senha*; e (ix) *saldo da conta*, de modo que seja possível:
- cadastrar dinamicamente (lembrar-se do uso dos operadores *new* e *delete*) *n* clientes, armazenando seus dados em arquivo denominado *cad_cliente.txt*, o qual deverá conter no mínimo 5 usuários;
 - autenticar um cliente cadastrado via tipo booleano, a partir de uma função-membro que verifique se o cliente existe ou não;
 - informar a um cliente cadastrado o saldo de sua conta; e
 - exibir na tela mensagens indicando a razão pela qual um usuário cadastrado não conseguiu realizar o saque, caso isto aconteça, assim como de uma mensagem informando que o saque foi realizado com sucesso. **1,3**

ÊXITO!