MAC 110/115 – Introdução à Computação – 1^o semestre de 2008
2^a Prova – 26/06/2008

NOME: _____
ASSINATURA: _____ NUSP: _____

Recomendações Gerais:

1. A prova é individual e com consulta vedada a apontamentos e colegas;
2. Duração da prova = uma hora e trinta minutos;
3. Conteúdo da prova = quatro questões e esta página de rosto – verifique antes do início da prova se seu caderno de questões está completo;
4. Use o verso das páginas se necessário; não podem ser utilizadas folhas avulsas e as folhas desse caderno não podem ser destacadas;
5. Somente serão consideradas soluções apresentadas com clareza e organização; Indique claramente os trechos de rascunho (que não devem ser corrigidos);
6. Resolver uma questão quer dizer:
   - escrever a classe;
   - escrever os métodos e atributos que compõem a classe
7. Pode-se (deve-se) resolver as questões a lápis.
8. **VÁ BEM NESSA PROVA !!!**

questão	valor	nota
1	2.5	
2	2.5	
3	2.5	
4	2.5	

Questão 1. Considere a seguinte interface:

```
interface AlunoDeIntroduçãoCC
{
   String nome();
   double médiaFinal();
   String resultadoFinal();
}
```

Estes métodos devolvem respectivamente o nome do aluno, sua média final em Introdução à Computação e o resultado final (aprovado, recuperação ou reprovado). Você deve construir duas classes `AlunoIntroduçãoCCIME` e `AlunoIntroduçãoCCUniTabajara` que implementam esta interface, de tal forma que:

(a) a classe `AlunoIntroduçãoCCIME` possui um construtor que recebe o nome do aluno, sua média de provas e sua média de EPs. O cálculo da média final depende das médias de provas e EPs: se ambos as médias forem maiores ou iguais a 5.0, a média final é computada dando peso 3 para média de provas e 1 para média de EPs; caso contrário, a média final é o mínimo destes valores. O resultado final será: aprovado, se a média final for pelo menos 5.0; recuperação, se a nota for pelo menos 3.0 e menor que 5.0; ou reprovado, caso contrário.

(b) a classe `AlunoIntroduçãoCCUniTabajara` possui um construtor que recebe o nome do aluno, a média de provas e um `boolean` indicando se o aluno foi aprovado pelo conselho de professores. A média final é normalmente igual à média de provas, mas será `max`(média de provas,7.0) se o aluno foi aprovado pelo conselho de professores. O resultado final é aprovado, se a média final for pelo menos 7.0, ou reprovado, caso contrário.

Questão 2. Considere a interface definida na questão anterior. Considere que é dada (**não precisa implementar!!!**) a classe `LeitorDeAlunos`, cuja interface é a seguinte:

```
class LeitorDeAlunos
{
    public AlunoDeIntroduçãoCC[] lêTurma( int númeroDeAlunosNaTurma );
}
```

Crie uma classe `AvaliadorDeTurma` com quantos atributos achar necessário, com os seguintes métodos:

(a) um construtor que recebe o número de alunos da turma, e usa o método da classe `LeitorDeAlunos` para obter os dados dos alunos.

(b) um método `double médiaDaTurma()`, que devolve a média da classe (média das médias finais de todos os alunos).

(c) um método `boolean éTurmaBoa()`, que devolve `true` se a percentagem de alunos aprovados é superior a 80% e `false` caso contrário.

(d) um método `void melhoresAlunos()` que imprime na tela o nome de todos os alunos com média final estritamente maior que a média da turma.

Questão 3. Implemente uma classe para representar figuras através de matrizes de caracteres formadas pelos caracteres `' '` (espaço) e `'X'`. Você pode utilizar os atributos que quiser. Sua classe deve conter os seguintes métodos:

(a) um construtor que recebe as dimensões da matriz e inicializa a figura com espaços.

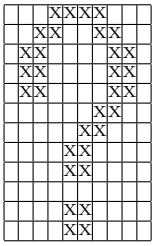
(b) um método `void imprime()` que mostra a matriz na tela.

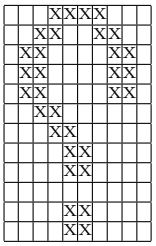
(c) um método `void muda(i,j)` que troca o caractere da posição `[i][j]` de `' '` para `'X'` e vice-versa. Atenção: seu método deve verificar se os valores de `i` e `j` estão na faixa correta, e ignorar a mensagem se esta condição não se verificar.

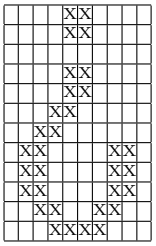
(d) um método `void inverte()` que troca todos os `' '` por `'X'` e vice-versa.

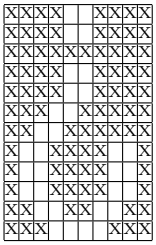
(e) um método `void espelha(char eixo)` que espelha a figura na horizontal se (eixo=='H') e na vertical se `eixo=='V'` (a mensagem não deve fazer nada se o valor de eixo for outro).

Exemplo: figura original e seqüência obtida após espelha(V), espelha(H) e inverte():









Questão 4. Considere o código de ordenação pelo método bolha implementado abaixo:

```
static void ordenaçãoBolha(int[] v)
{
    int i;                // índice usado para percorrer o vetor
    boolean estáOrdenado; // usado para checar se precisa
                          // repetir a varredura do vetor

    estáOrdenado = false;
    while (!estáOrdenado)
    {
        for (i=0;i<v.length-1;i=i+1)
        {
            if (v[i]>v[i+1])
            {
                v[i]=v[i+1];
                v[i+1]=v[i];
                estáOrdenado=false;
            }
            else estáOrdenado=true;
        }
    }
}
```

O programa acima apresenta alguns erros.

(a) mostre uma entrada (de preferência pequena) em que o método não funciona, explicitando o conteúdo do vetor antes e depois da chamada `ordenaçãoBolha(v)`.

(b) apresente uma versão corrigida do método.