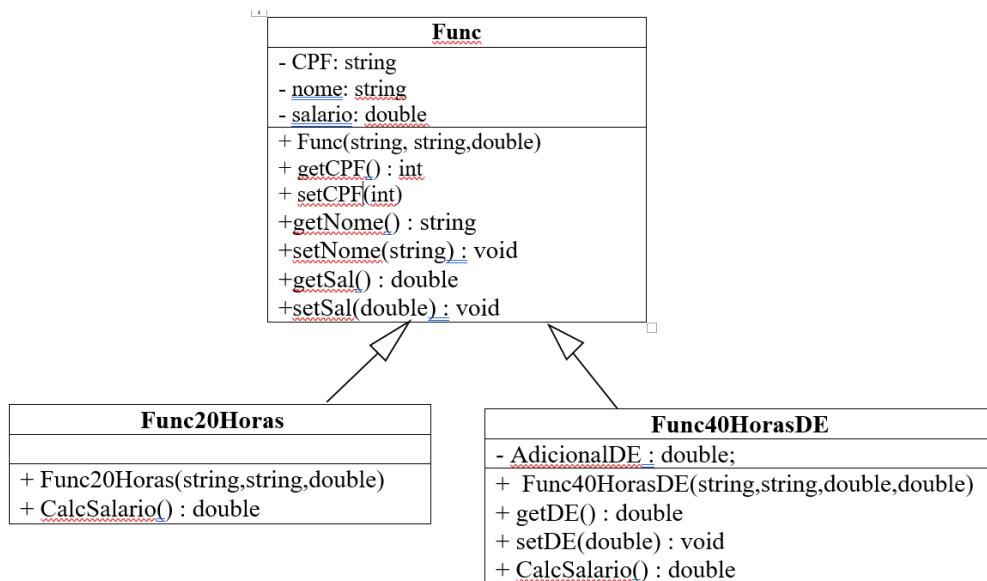


UNIVERSIDADE FEDERAL DE VIÇOSA
DEPARTAMENTO DE INFORMÁTICA
Inf 213 - Estrutura de dados
Primeira prova – 30/03/17
Prof. Marcus Vinícius Alvim Andrade

1) (40%) a) Declare as classes definidas no diagrama UML ao lado que representam as categorias de funcionários de uma empresa: funcionário de 20 horas e funcionário de 40 horas com dedicação exclusiva (DE). **NÃO PRECISA** implementar as funções membros.

b) Implemente a função *CalcSalario* das classes *Func20Horas* e *Func40HorasDE* que devem ter o seguinte comportamento: para o funcionário de 20 horas o valor do salário é o valor que está armazenado no atributo *salario*; para o funcionário 40 horas com DE, o valor é a soma dos atributos *salario* e *AdicionalDE*.



c) Escreva uma função (não membro das classes) que recebe como parâmetro um array contendo todos os funcionários de uma empresa e o tamanho do array e imprime o CPF e o salário de todos os funcionários.

d) Escreva uma função (não membro das classes) que recebe como parâmetro um array contendo todos os funcionários de uma empresa e o tamanho do array e imprime os dados (o CPF, o nome e o salário) do funcionário com o maior salário. Além disso, a função deve imprimir qual a categoria (*Funcionario20Horas* ou *Funcionario40HorasDE*) deste funcionário com o maior salário.

2) (30%) Dada a classe Q2 e a função *f* abaixo, mostre o resultado dos trechos de código. Se houver algum erro (compilação ou execução) explique a razão do erro e faça as correções necessárias **NA CLASSE** e depois liste o resultado. **OBS:** você **só pode** alterar a classe.

```
template <class T>
class Q2 {
public:
    Q2(int n) { ptr = new T[n]; dim=n;}
    ~Q2() { cout << "Desaloca ptr \n"; delete [] ptr;}
    void set(T x, int i) {ptr[i]=x; }
    T get(int i) { return ptr[i];}

private:
    T *ptr;
    int dim;
};
```

```
template <class T>
void f(Q2<T> u) {
    cout << u.get(0) << endl;
}
```

```
a) Q2<char> a;
   a.set('A',0);
   cout << a.get(0) << endl;

b) Q2<char> b(5);
   b.set('B',0);
   {
       Q2<char> x = b;
       cout << x.get(0) << endl;
   }
   cout << b.get(0) << endl;
```

```
c) Q2<int> c(4);
   c.set(2,0);
   f(c);
   int i = c.get(0);
```

CONTINUA NO VERSO!!!

3) (30%) Considerando as declarações as classes *P2* e *P3* abaixo, verifique se os trechos de códigos dados nas letras a), b), c) e d) podem ser compilados e, em caso afirmativo, escreva o resultado obtido. Caso o trecho não possa ser compilado, indique a causa do erro.

```
class P2 {
public:
    P2() { x = y = 0; }
    void set(double xi, double yi) { x = xi; y = yi; }
}
    virtual double calc() { return (x*y); }
    void print() { cout << x << " " << y; }
protected:
    double x,y;
};
```

```
class P3 : public P2 {
public:
    P3() : P2() { z = 0; c = '\0'; }
    void set(double xi, double yi, double zi=0, char ci='#')
        { x = xi; y = yi; z = zi; c = ci; }
    virtual double calc() { return ((x+y) / z); }
    virtual void print() { P2::print(); cout << " " << z << " " << c; }
protected:
    double z;
    char c;
};
```

a)

```
P2 m;
P3 n;
m.set(5,10);
n.set(4,3,2,'%');
n = m;
cout << "m = "; m.print();
cout << " -- m.calc = " << m.calc() << endl;
cout << "n = "; n.print();
cout << " -- n.calc = " << n.calc() << endl;
cout << endl;
```

b)

```
P2 p;
P3 q;
P2 *r;
p.set(5,10);
q.set(3,2,2,'%');
r = &q;
cout << "p = "; p.print();
cout << " -- p.calc = " << p.calc() << endl;
cout << "q = "; q.print();
cout << " -- q.calc = " << q.calc() << endl;
cout << "r = "; r->print();
cout << " -- r.calc = " << r->calc() << endl;
p = q;
r = &p;
cout << "p = "; p.print();
cout << " -- p.calc = " << p.calc() << endl;
cout << "q = "; q.print();
cout << " -- q.calc = " << q.calc() << endl;
cout << "r = "; (*r).print();
cout << " -- r.calc = " << (*r).calc() << endl;
cout << endl;
```

c)

```
P2 *a = new P2;
P3 *b = new P3;
a->set(10,20);
b->set(4,3,2);
cout << "a = "; a->print();
cout << " -- a.calc = " << a->calc() << endl;
cout << "b = "; b->print();
cout << " -- b.calc = " << b->calc() << endl;
a = b;
cout << "a = "; a->print();
cout << " -- a.calc = " << a->calc() << endl;
cout << "b = "; b->print();
cout << " -- b.calc = " << b->calc() << endl;
```

d)

```
P2 *m2 = new P2;
P3 *m3 = new P3;
m2->set(10,20);
m3->set(4,3,2);
m3 = m2;
cout << "m2 = "; m2->print();
cout << " -- m2.calc = " << m2->calc() << endl;
cout << "m3 = "; m3->print();
cout << " -- m3.calc = " << m3->calc() << endl;
```