Engenharia Informática/Informática de Gestão
Época Normal – 19 de janeiro de 2020

Grupo I (15 valores)

a) Defina em C++ todas as classes do problema, respeitando integralmente os métodos ... [11.4 val.]

```
class Inventario{
      ColecaoHibrida<Sala*> salas;    4
public:

      bool addSala(int n) {           2
            Sala *s = new Sala(n);
            return salas.insert(s);
      }
      bool addSalaAula(int n, int numL, double valL) {   2
            Sala *sa = new SAula(n, numL, valL);
            return salas.insert(sa);
      }
      bool registarEquipamento(int sala, double val) {   3
            Sala *s = findSala(sala);
            if (s != NULL)
                  return s->addEquipamento(val);
            else return false;
      }
      double valorTotal() {                              3
            double tot = 0.0;
            ColecaoHibrida<Sala*>::iterator it;
            for (it = salas.begin(); it != salas.end(); it++)
                  tot += (*it)->valor();
            return tot;
      }
private:
      Sala *findSala(int n){          2
            Sala  s(n);
            return salas.find(&s);
      }
};

class Equipamento{          1
      int cod;
      double val;
public:
      Equipamento(int c, double v){     1
            cod = c;  val = v;
      }
      double getValor() const { return val;}   1

      bool operator<(const Equipamento &outro)const {   1
            return cod < outro.cod;
      }
};
```

16

4

```
class Sala{                                                    2                    11
      int num;
      Colecao<Equipamento> equipamentos;
public:
      Sala(int n) { num = n; }   1

      virtual double valor() {                  4
            double tot=0.0;
            Colecao<Equipamento>::iterator it;
            for (it = equipamentos.begin(); it != equipamentos.end(); it++)
                  tot += it->getValor();
            return tot;
      }
      bool addEquipamento(double v) {                    3
            int cod = equipamentos.size() + 1;
            Equipamento e(cod, v);
            return equipamentos.insert(e);
      }7
      bool operator<(const Sala &outra)const {      1
            return num < outra.num;
      }
};

class SAula: public Sala {       2                                              7
      int numLugares;
      double valLugar;
public:
      SAula(int n, int numL, double valL):Sala(n){numLugares=numL; valLugar=valL;}2
      double valor() {return Sala::valor() + numLugares * valLugar;}   3
};
```

b) Acrescente ao problema os métodos que permitam apresentar na saída .... [3.6 val.]

```
bool Inventario::listarSala(int sala) {   3                                       12
      Sala *s = findSala(sala);
      if (s != NULL) {
            s->listar();
            return true;
      }
      else return false;
}
virtual void Sala::listar() const{       4
      Colecao<Equipamento>::iterator it;
      for (it = equipamentos.begin(); it != equipamentos.end(); it++)
            it->print();
}
void SAula::listar() const{      3
      Sala::listar();
      cout<<"<Equipamento dos lugares> Valor estimado: "<<numLugares*valLugar
            << " euros." << endl;
}
void Equipamento::print() const{      2
      cout<<"Equipamento "<<cod<< " Valor estimado: " << val << " euros." << endl;
}
```

Grupo II (5 valores)

a) Diga quais são os métodos construtores (definidos explicitamente ou não) que... [0.6 val.]

Nas classes A e C estão presentes quer o construtor de conversão quer o de cópia;
Já na classe B estão presentes o construtor por defeito e o de cópia.

b) Apresente o resultado que será visualizado na saída standard com a execução do programa. [2.8 val.]

```
Criado A(1)                    2
Criado B                       1
Criado A(2)                    2
Criado B                       1
Criado A(3)                    2
Criado B                       1
Criado um C com 3 Bs           4 (2 se na pos. errada)
-1-
C:B:A(1)                       2
C:B:A(2)                       2
C:B:A(3)                       2
-2-
Destruido C                    3 (1 se na pos. errada)
Destruido B                    1
Destruido A                    1
Destruido B                    1
Destruido A                    1
Destruido B                    1
Destruido A                    1
```

c) Diga que consequências é que teria, naquilo que é visualizado, o método `print()` ... [0.4 val.]

Nada se alteraria (estamos na presença de relações de agregação, não de herança).

d) Dê uma breve explicação do erro cometido quando se acrescenta ao `main` do programa: [1.2 val.]

(1) `cout << objeto << endl;`

O operador insersor (<<) não se encontra definido para operandos do tipo C.

(2) `A vetor[3];`

A classe A não dispõe de construtor por defeito.

(3) `A *p = new B();`

Está-se a tentar fazer uma conversão ascendente entre apontadores, só possível, se B derivasse de A.

(4) `C *p = new C(objeto); delete p; objeto.print();`

É criado dinamicamente um objeto da classe C, por cópia de um outro que já existe, ficando os dois objetos a partilhar o mesmo vetor de objetos B; esse array é depois eliminado com a eliminação do objeto criado; logo, o método print() do objeto inicial vai tentar aceder a um array que já não existe.