

Um escritório possui múltiplas impressoras distribuídas pelas suas salas. Pretende-se monitorar o uso dessas impressoras, nomeadamente o consumo de toner.

A informação sobre uma impressora inclui um código identificador, o ano de aquisição e os documentos já impressos (vetor). Todas as impressoras usam toner. Existem impressoras a preto e branco e impressoras a cores:

- O consumo de toner da impressora a preto e branco é igual a NumPagDoc (NumPagDoc é o número de páginas do documento a imprimir). O membro-dado *numPagImprimir* especifica o número de páginas que ainda é possível imprimir com a quantidade de toner existente.
- O consumo de toner da impressora a cores, é calculado para cada um dos toners (por uma questão de simplificação, considera-se a existência de apenas duas cores: preto, amarelo). O consumo de toner de uma cor (preto, amarelo) é igual a $\text{numPaginas} * \text{pCor}$, em que *numPaginas* é o número de páginas do documento a imprimir e *pCor* é a percentagem dessa cor no documento.

A classe Documento identifica o documento a ser impresso. Contém informação sobre o número de páginas deste (*numPaginas*) e sobre a percentagem de cada cor - preto, amarelo - presente no documento (*pPreto*, *pAmarelo*). Esta percentagem é um valor entre 0 e 1.

O escritório possui vários funcionários, cada um responsável por um conjunto de impressoras.

As classes **Escritorio**, **Documento**, **Funcionario**, **Impressora**, **ImpressoraPB**, **ImpressoraCores** estão parcialmente definidas a seguir.

NÃO PODE acrescentar membros-dado nas classes **Impressora**, **ImpressoraPB** e **ImpressoraCores**.

```
class Impressora {
protected:
    const string codigo;
    const tint ano;
    vector<Documento> docsImpressos;
public:
    Impressora(string cod, int a);
    // ...
};

class ImpressoraPB: public Impressora {
    int numPagImprimir;
public:
    ImpressoraPB(string cod, int a,
                  int toner);
    // ...
};

class ImpressoraCores: public Impressora {
    float numPagImprimirPreto;
    float numPagImprimirAmarelo;
public:
    ImpressoraCores(string cod, int a,
                    int toner);
    // ...
};

class Escritorio {
    vector<Impressora *> impressoras;
    vector<Funcionario> funcionarios;
public:
    Escritorio();
    void adicionaImpressora(Impressora *i1);
    void adicionaFuncionario(Funcionario f1);
    // ...
};

class Documento {
    int numPaginas;
    float pPreto, pAmarelo;
public:
    Documento(int nPag, float pP, float
              pA);
    // ...
};

class Funcionario {
    const string codigo;
    vector<Impressora *> impressoras;
public:
    Funcionario(string cod);
    void adicionaImpressora(Impressora *i1);
    // ...
};
```

- a) [2.5 valores] Implemente na classe **Escritorio** o membro-função:

int numImpressorasSemResponsavel() const

Esta função retorna o número de impressoras que não têm responsável. Um funcionário pode ser responsável por múltiplas impressoras, mas uma impressora tem, no máximo, um responsável.

- b) [3 valores] Implemente na classe **Escritorio** o membro-função:

*vector<Impressora *> retiraImpressoras(int ano1)*

Esta função remove do vetor `impressoras`, todas as impressoras com ano de aquisição menor que `ano1`. Retorna um vetor com (referências para) as impressoras removidas.

- c) [3.5 valores] A impressão de um documento só é possível se a impressora tiver toner. Implemente, nas classes que considerar necessário, o membro-função que efetua a impressão de um documento:

bool imprime(Documento doc1)

Esta função retorna `true`, se a impressão pode ser realizada e `false` senão. No caso da impressão ser realizada, a função deve ainda atualizar os membros-dados que indicam o número de páginas possível de imprimir (após o gasto de toner).

Numa impressora a preto e branco:

`numPagImprimir` é decrementado de um valor igual a nº de páginas do documento

Numa impressora a cores:

`numPagImprimirPreto` é decrementado de um valor igual a nº páginas documento * percentagem de cor preto do documento

`numPagImprimirAmarelo` é decrementado de um valor igual a nº páginas documento * percentagem de cor amarelo do documento

A função deve adicionar o documento ao vetor `docsImpressos`.

- d) [2.5 valores] Implemente na classe **Escritorio** o membro-função:

*Impressora * imprimeDoc(Documento doc1) const*

Esta função efetua a impressão (de acordo com o enunciado na alínea anterior) do documento `doc1` na primeira impressora capaz de executar a impressão. Por primeira entende-se a ordem relativa à posição no vetor `impressoras`. A função retorna uma referência para a impressora que efetuou a impressão. Se não existe nenhuma impressora capaz de imprimir `doc1`, a função retorna a referência para uma `Impressora` de código igual a "inexistente".

- e) [3 valores]] Implemente na classe **Escritorio** o membro-função:

*vector<Impressora *> tonerBaixo() const;*

Esta função funciona como alerta para a manutenção das impressoras e retorna um vetor de (referências para as) impressoras com nível de toner baixo. Uma impressora a preto e branco tem toner baixo se o número de páginas que é possível imprimir é menor que 30. Uma impressora a cores tem toner baixo se o número de páginas que é possível imprimir, em pelo menos uma cor, é menor que 20.

- f) [2.5 valores] Implemente na classe **Documento** o operador `+`. A soma de dois documentos `d1` e `d2` é um documento com número de páginas igual à soma das páginas de `d1` e `d2`. A percentagem de cada

cor é igual a $(\text{percentagemCor_d1} * \text{numPag_d1} + \text{percentagemCor_d2} * \text{numPag_d2}) / (\text{numPag_d1} + \text{numPag_d2})$

- g) [3 valores] Implemente na classe **Escritorio** o operador de função que aceita um argumento do tipo *string* (representando o código de uma impressora) e retorna o código do funcionário responsável. Se não existir nenhuma impressora com o código indicado ou a impressora não tem nenhum funcionário alocado, retorna a string “nulo”.