

== Laboratório 4 ==

1. Escreva um programa que utilize uma classe `Agenda`. Essa classe possui um objeto da classe `vector` como atributo privado, que é utilizado para armazenar `N` nomes inseridos pelos usuários. Para tal, a classe `Agenda` deve oferecer um método para inserção de nomes. Os nomes devem ter no máximo 10 caracteres. Caso o nome possua mais que 10 caracteres, uma mensagem deve ser exibida ao usuário e o nome deve ser truncado para caber no limite máximo definido. Após a inserção dos nomes, o usuário deve ser capaz de visualizar todos os nomes através de um método da classe `Agenda`. Para truncar os nomes utilize a função (`string substr (size_t pos = 0, size_t n = npos) const;`) definida na biblioteca `string`. O número `N` de nomes deve ser passado para o construtor da classe `Agenda` como um argumento padrão. Esse valor é usado para inicializar um atributo privado que irá controlar o número máximo de nomes na agenda. Use o método `push_back` para inserir novos elementos em um `vector` e o método `size` para saber o número de elementos já inseridos no `vector`.
2. Reescreva o programa anterior, substituindo o `vector` por um array de strings. Utilize um atributo constante estático para estabelecer o tamanho do array e use um atributo estático para controlar o número de elementos já inseridos no array. Veja se ainda é possível receber o número máximo de elementos do array pelo construtor.
3. Reescreva novamente o Exercício 1, substituindo o array de strings por um array de objetos da classe `Cadastro`. A classe `Cadastro` possui como atributos privados o nome, a profissão e a idade de cada uma das pessoas registradas. Além dos atributos, essa classe possui métodos do tipo “get” e “set” para cada um dos seus atributos. O método `insertName` da classe `Agenda` é modificado para `insertCad`, já que recebe uma referência a um objeto da classe `Cadastro`, no lugar de apenas um nome. Tal objeto foi previamente inicializado através de seus métodos `set`. O método `showNames` foi também alterado para exibir na tela todos os atributos da classe `Cadastro`.
4. Reescreva o programa anterior, substituindo o array de objetos da classe `Cadastro` por um `vector` de objetos da classe `Cadastro`. Realize as outras alterações necessárias para que o programa funcione como o programa do exercício anterior.

== Respostas ==

1.

```
/*  
*****  
***** Programa Principal *****  
*/  
  
#include <iostream>  
#include <string>  
#include <vector>
```

```

#include "agenda.h"

using namespace std;

int main () {
    Agenda agenda;
    string name;

    cout << "Entre com três nomes: " << endl;
    cout << endl;

    for (int i = 0; i < 3; i++) {
        cout << "Nome [" << i << "]: ";
        getline (cin, name);
        agenda.insertName (name);
    }

    // Mostrar os nomes
    agenda.showNames ();

    return 0;
}

/*****
/***** Arquivo agenda.h *****/

#include <iostream>
#include <string>
#include <vector>

using namespace std;

class Agenda {
public:
    Agenda (int = 3);
    void insertName (string);
    void showNames ();

private:
    vector <string> v;
    int maxNames;

    string verifyName (string);
};

/*****
/***** Arquivo agenda.cpp *****/

#include "agenda.h"

Agenda::Agenda (int n) {
    maxNames = n;
}

void Agenda::insertName (string str) {
    // verifyName trunca o nome str caso necessário
    str = verifyName (str);

    if (v.size () > maxNames)
        cout << "Agenda cheia!" << endl;
    else
        v.push_back(str);
}

void Agenda::showNames () {
    for (int i = 0; i < v.size (); i++) {
        cout << "Nome [" << i << "] "
            << v.at (i) << endl;
    }
}

string Agenda::verifyName (string str) {
    if (str.length () > 10) {
        cout << "Nome com mais de 10 caracteres, reduzindo para 10...";
        cout << endl;
        str = str.substr (0, 10);
    }
}

```

```

    }
    return str;
}

/*****/

```

## 2.

```

/*****/
/***** Programa Principal *****/

#include <iostream>
#include <string>

#include "agenda.h"

using namespace std;

int main () {
    Agenda agenda;
    string name;

    cout << "Entre com três nomes: " << endl;
    cout << endl;

    for (int i = 0; i < 3; i++) {
        cout << "Nome [" << i << "]: ";
        getline (cin, name);
        agenda.insertName (name);
    }

    // Mostrar os nomes
    agenda.showNames ();

    return 0;
}

/*****/
/***** Arquivo agenda.h *****/

#include <iostream>
#include <string>

using namespace std;

class Agenda {
public:
    void insertName (string);
    void showNames ();

private:
    const static int maxNames = 3;
    string names [maxNames];
    static int numberOfNames;

    string verifyName (string);
};

/*****/
/***** Arquivo agenda.cpp *****/

#include "agenda.h"

int Agenda::numberOfNames = 0;

void Agenda::insertName (string str) {
    str = verifyName (str);

    if (numberOfNames >= maxNames)
        cout << "Agenda cheia!" << endl;
    else
        names [numberOfNames++] = str;
}

void Agenda::showNames () {

```

```

        for (int i = 0; i < numberOfNames; i++) {
            cout << "Nome [" << i << "] "
                 << names [i] << endl;
        }
    }

string Agenda::verifyName (string str) {
    if (str.length () > 10) {
        cout << "Nome com mais de 10 caracteres, reduzindo para 10...";
        cout << endl;
        str = str.substr (0, 10);
    }
    return str;
}

/*****

```

### 3.

```

/*****
/***** Programa Principal *****/

#include <iostream>
#include <string>

using namespace std;

#include "cadastro.h"
#include "agenda.h"

int main () {
    Agenda agenda;
    string name, profissao;
    int idade;

    cout << "Entre com três nomes: " << endl;
    cout << endl;

    for (int i = 0; i < 3; i++) {
        Cadastro cad;

        cout << "Nome [" << i << "]: ";
        getline (cin, name);
        cad.setName (name);

        cout << "Profissao [" << i << "]: ";
        getline (cin, profissao);
        cad.setProf (profissao);

        cout << "Idade [" << i << "]: ";
        cin >> idade;
        cin.ignore ();
        cad.setAge (idade);

        agenda.insertCad (cad);
    }

    // Mostrar os nomes
    agenda.showCads ();

    return 0;
}

/*****
/***** Arquivo agenda.h *****/

#include <iostream>
#include <string>
#include <vector>

using namespace std;

class Agenda {
public:
    void insertCad (Cadastro &);

```

```

        void showCads ();

    private:
        const static int maxNames = 3;
        Cadastro names [maxNames];
        static int numberOfNames;
};

/*****
/***** Arquivo agenda.cpp *****/

#include "agenda.h"

int Agenda::numberOfNames = 0;

void Agenda::insertCad (Cadastro &c) {
    if (numberOfNames >= maxNames)
        cout << "Agenda cheia!" << endl;
    else
        names [numberOfNames++] = c;
}

void Agenda::showCads () {
    for (int i = 0; i < numberOfNames; i++) {
        cout << "Nome [" << i << "] " << names [i].getName () << endl
            << "Prof [" << i << "] " << names [i].getProf () << endl
            << "Idade [" << i << "] " << names [i].getAge () << endl;
    }
}

/*****
/***** Arquivo cadastro.h *****/

#include <iostream>
#include <string>
#include <vector>

using namespace std;

class Cadastro {
public:
    void setName (string);
    string getName ();

    void setProf (string);
    string getProf ();

    void setAge (int);
    int getAge ();

private:
    string name, prof;
    int age;

    string verifyName (string);
};

/*****
/***** Arquivo cadastro.cpp *****/

#include "cadastro.h"

void Cadastro::setName (string n) {
    n = verifyName (n);
    name = n;
}

string Cadastro::getName () {
    return name;
}

void Cadastro::setProf (string p) {
    prof = p;
}

string Cadastro::getProf () {

```

```

        return prof;
    }

void Cadastro::setAge (int a) {
    age = a;
}

int Cadastro::getAge () {
    return age;
}

string Cadastro::verifyName (string str) {
    if (str.length () > 10) {
        cout << "Nome com mais de 10 caracteres, reduzindo para 10...";
        cout << endl;
        str = str.substr (0, 10);
    }
    return str;
}

/*****

```

#### 4.

```

/*****
/***** Programa Principal *****/

#include <iostream>
#include <string>
#include "cadastro.h"
#include "agenda.h"

using namespace std;

int main () {
    Agenda agenda;
    string name, profissao;
    int idade;

    cout << "Entre com três nomes: " << endl;
    cout << endl;

    for (int i = 0; i < 3; i++) {
        Cadastro cad;

        cout << "Nome [" << i << "]: ";
        getline (cin, name);
        cad.setName (name);

        cout << "Profissao [" << i << "]: ";
        getline (cin, profissao);
        cad.setProf (profissao);

        cout << "Idade [" << i << "]: ";
        cin >> idade;
        cin.ignore ();
        cad.setAge (idade);

        agenda.insertCad (cad);
    }

    // Mostrar os nomes
    agenda.showCads ();

    return 0;
}

/*****
/***** Arquivo agenda.h *****/

#include <iostream>
#include <string>
#include <vector>

using namespace std;

```

```

class Agenda {
public:
    void insertCad (Cadastro &);
    void showCads ();

private:
    const static int maxNames = 3;
    vector <Cadastro> v;
};

/*****
/***** Arquivo agenda.cpp *****/

#include "agenda.h"

void Agenda::insertCad (Cadastro &c) {
    if (v.size () > maxNames)
        cout << "Agenda cheia!" << endl;
    else
        v.push_back (c);
}

void Agenda::showCads () {
    for (int i = 0; i < v.size (); i++) {
        cout << "Nome [" << i << "] " << v [i].getName () << endl
            << "Prof [" << i << "] " << v [i].getProf () << endl
            << "Idade [" << i << "] " << v [i].getAge () << endl;
    }
}

/*****
/***** Arquivo cadastro.h *****/

#include <iostream>
#include <string>
#include <vector>

using namespace std;

class Cadastro {
public:
    void setName (string);
    string getName ();

    void setProf (string);
    string getProf ();

    void setAge (int);
    int getAge ();

private:
    string name, prof;
    int age;

    string verifyName (string);
};

/*****
/***** Arquivo cadastro.cpp *****/

#include "cadastro.h"

void Cadastro::setName (string n) {
    n = verifyName (n);
    name = n;
}

string Cadastro::getName () {
    return name;
}

void Cadastro::setProf (string p) {
    prof = p;
}

```

```
string Cadastro::getProf () {
    return prof;
}

void Cadastro::setAge (int a) {
    age = a;
}

int Cadastro::getAge () {
    return age;
}

string Cadastro::verifyName (string str) {
    if (str.length () > 10) {
        cout << "Nome com mais de 10 caracteres, reduzindo para 10...";
        cout << endl;
        str = str.substr (0, 10);
    }
    return str;
}

/*****/
```