
Laboratorio 17

Programación genérica

Nombre: Gabriel Fernando Rodriguez Cutimbo

CUI: 20212157

Grupo: B

Repositorio GitHub:

https://github.com/gaco123/EPCC_CCII.git

1. Ejercicios

Resolver los siguientes ejercicios planteados:

1. Desarrolle un programa simple de calculadora (operaciones basicas) que utilice clases utilizando plantillas

Código:

```
/*  
    1. Desarrolle un programa simple de calculadora (operaciones basicas) que utilice  
    clases utilizando plantillas  
*/  
#include <iostream>  
using namespace std;  
  
template <class X, class Y>  
class op_basica{  
public:  
    op_basica(int op, X a, Y b){  
        if(op==1){  
            cout<<"La suma de los números ingresados es: "<<a+b;  
        }  
        else if(op==2){  
            cout<<"La resta de los números ingresados es: "<<a-b;  
        }  
        else if(op==3){  
            cout<<"La multiplicación de los números ingresados es: "<<a*b;  
        }  
        else if(op==4){
```

```
        cout<<"La división de los números ingresados es: "<<a/b;
    }
}
};

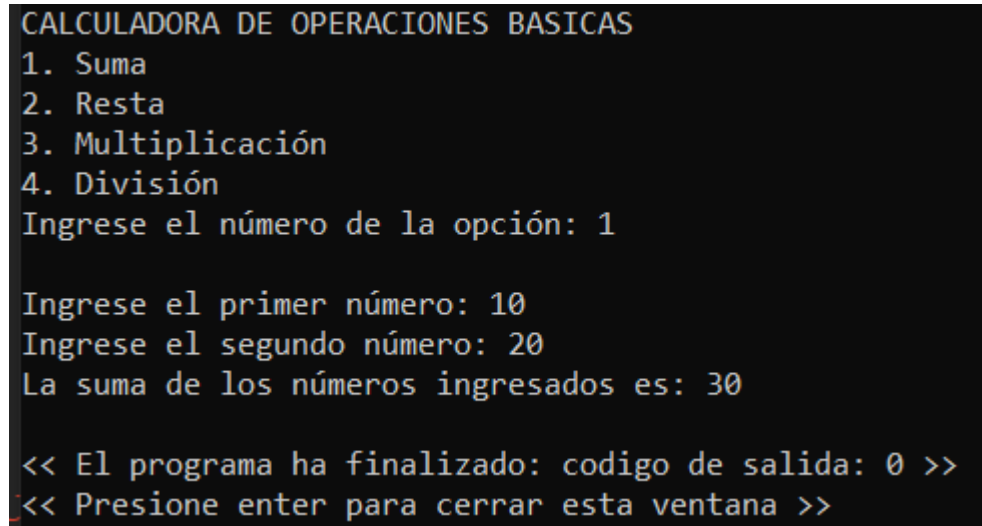
int main() {
    int op;

    do{
        system("CLS");
        cout<<"CALCULADORA DE OPERACIONES BASICAS\n";
        cout<<"1. Suma\n";
        cout<<"2. Resta\n";
        cout<<"3. Multiplicación\n";
        cout<<"4. División\n";
        cout<<"Ingrese el número de la opción: ";
        cin>>op;
        if(op==1||op==2||op==3||op==4){
            break;
        }
        else{
            cout<<"Error ingrese de nuevo la opción...\n\n";
            system("PAUSE");
        }
    }while(true);
    cout<<endl;

    if(op==1){
        int a, b;
        cout<<"Ingrese el primer número: ";
        cin>>a;
        cout<<"Ingrese el segundo número: ";
        cin>>b;
        op_basica<int,int> x(1,a,b);
    }
    else if(op==2){
        float a, b;
        cout<<"Ingrese el primer número: ";
        cin>>a;
        cout<<"Ingrese el segundo número: ";
        cin>>b;
        op_basica<float,float> x(2,a,b);
    }
    else if(op==3){
        long a, b;
        cout<<"Ingrese el primer número: ";
        cin>>a;
        cout<<"Ingrese el segundo número: ";
        cin>>b;
        op_basica<long,long> x(3,a,b);
    }
    else if(op==4){
        double a, b;
        cout<<"Ingrese el primer número: ";
        cin>>a;
        cout<<"Ingrese el segundo número: ";
```

```
        cin>>b;
        op_basica<double,double> x(4,a,b);
    }

    return 0;
}
```

Funcionamiento:

```
CALCULADORA DE OPERACIONES BASICAS
1. Suma
2. Resta
3. Multiplicación
4. División
Ingrese el número de la opción: 1

Ingrese el primer número: 10
Ingrese el segundo número: 20
La suma de los números ingresados es: 30

<< El programa ha finalizado: codigo de salida: 0 >>
<< Presione enter para cerrar esta ventana >>
```

2. Definir una clase utilizando plantillas que permita almacenar datos en un arbol binario. Por el momento solo se insertaran elementos en la estructura. Simule el proceso de almacenar 100 datos y verifique que la estructura no tenga problemas.

Código:

```
/*
2. Definir una clase utilizando plantillas que permita almacenar datos en un arbol
binario. Por el momento solo se insertaran elementos en la estructura. Simule el
proceso de almacenar 100 datos y verifique que la estructura no tenga problemas.
*/

#include <iostream>
using namespace std;

template <class X>
class nodo{
public:
    nodo(X val){
        this->val = val;
        this->sig = nullptr;
    }
}
```

```
X val;
nodo<X>* sig;
};

template <class X>
class l_enlazada{
private:
    nodo<X>* head;
public:
    l_enlazada(X val){
        head = new nodo<X>(val);
    }
    ~l_enlazada(){
        nodo<X>* temp = head;
        while(temp->sig!=nullptr){
            head=temp->sig;
            delete temp;
            temp=head;
        }
        delete head;
    }
    void push_back(X val){
        nodo<X>* new1 = new nodo<X>(val);
        nodo<X>* temp = head;
        while(temp->sig!=nullptr){
            temp=temp->sig;
        }
        temp->sig = new1;
    }
    void print(){
        nodo<X>* temp = head;
        cout<<"[";
        while(temp!=nullptr){
            if(temp->sig!=nullptr){
                cout<<temp->val;
                cout<<" ";
            }
            else{
                cout<<temp->val;
            }
            temp=temp->sig;
        }
        cout<<"]";
    }
};

int main(){
    //Lista enlazada que almacena floats
    float a = 0.5;
    l_enlazada<float> x(a);

    for(int i=1; i<100; i++){
        x.push_back(a+i);
    }
    cout<<"LISTA ENLAZADA [FLOAT]\n";
    x.print();
}
```

```
cout<<endl<<endl;

//Lista enlazada que almacena ints
l_enlazada<int> y(0);

for(int i=1; i<100; i++){
    y.push_back(i);
}
cout<<"LISTA ENLAZADA [INT]\n";
y.print();

return 0;
}
```

Funcionamiento:

```
LISTA ENLAZADA [FLOAT]
[0.5, 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5, 10.5, 11.5, 12.5, 13.5, 14.5, 15.5, 16.5, 17.5, 18.5, 19.5, 20.5, 21.5, 22.5,
23.5, 24.5, 25.5, 26.5, 27.5, 28.5, 29.5, 30.5, 31.5, 32.5, 33.5, 34.5, 35.5, 36.5, 37.5, 38.5, 39.5, 40.5, 41.5, 42.5, 43.5,
44.5, 45.5, 46.5, 47.5, 48.5, 49.5, 50.5, 51.5, 52.5, 53.5, 54.5, 55.5, 56.5, 57.5, 58.5, 59.5, 60.5, 61.5, 62.5, 63.5, 64.5,
65.5, 66.5, 67.5, 68.5, 69.5, 70.5, 71.5, 72.5, 73.5, 74.5, 75.5, 76.5, 77.5, 78.5, 79.5, 80.5, 81.5, 82.5, 83.5, 84.5, 85.5, 86.5,
87.5, 88.5, 89.5, 90.5, 91.5, 92.5, 93.5, 94.5, 95.5, 96.5, 97.5, 98.5, 99.5]

LISTA ENLAZADA [INT]
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65,
66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97,
98, 99]
```

3. Analice y describa el siguiente comportamiento

Código:

```
/*
3. Analice y describa el siguiente comportamiento
*/

#include <iostream>
using namespace std;

//Clase genérica Contenedor
template<class T>
class Contenedor{
    T elemento;
public:
    //Constructor genérico de la clase "Contenedor"
    Contenedor(T arg){
        elemento = arg;
    }
    //Suma 1 al "elemento" que esta en la clase "Contenedor"
    T add(){
        return ++elemento;
    }
};
```

```
/*
    Descripción específica para la clase "Contenedor" cuando el tipo de valor
    tomado por el template es <char>
*/
template <T>
class Contenedor<char>{
    char elemento;
public:
    //Constructor de la clase "Contenedor<char>"
    Contenedor(char arg){
        elemento = arg;
    }
    //Esta función se encarga de volver una letra minúscula en mayúscula
    char uppercase(){
        if((elemento >= 'a') && (elemento <= 'z')){
            /*
                Al restar este número a una letra minúscula hara que está retroceda 32 espacios
                quedando exactamente en su misma letra pero en mayúscula
            */
            elemento += 'A'-'a';
        }
        return elemento;
    }
};

int main(){
    //Crea una clase "Contenedor" genérica que acepta elementos de tipo <int>
    Contenedor<int> cint(5);
    //Crea un "Contenedor<char>" que tiene su propia descripción
    Contenedor<char> cchar('t');
    //Utiliza la función add de la clase "Contenedor" genérica
    cout<<cint.add()<<endl;
    //Utiliza la función uppercase que es propia de la clase "Contenedor<char>"
    cout<<cchar.uppercase()<<endl;

    return 0;
}
```

Funcionamiento:



2. Entregables

Al final estudiante deberá:

1. Compactar el código elaborado y subirlo al aula virtual de trabajo. Agregue sus datos personales como comentario en cada archivo de código elaborado.
2. Elaborar un documento que incluya tanto el código como capturas de pantalla de la ejecución del programa. Este documento debe de estar en formato PDF.
3. El nombre del archivo (comprimido como el documento PDF), será su LAB17_GRUPO_A/B/C_CUI_1erNOMBRE_1erAPELLIDO.

(Ejemplo: LAB17_GRUPO_A_2022123_PEDRO_VASQUEZ).

4. Debe remitir el documento ejecutable con el siguiente formato:

LAB17_GRUPO_A/B/C_CUI_EJECUTABLE_1erNOMBRE_1erAPELLIDO

(Ejemplo: LAB17_GRUPO_A_EJECUTABLE_2022123_PEDRO_VASQUEZ).

En caso de encontrarse trabajos similares, los alumnos involucrados no tendrán evaluación y serán sujetos a sanción.