CS1112: Programación II

Unidad 5: POO

Sesión de Laboratorio - 9A

Profesores:
María Hilda Bermejo **mbermejo@utec.edu.pe**
Estanislao Contreras **econtreras@utec.edu.pe**
Jorge Villavicencio **jvillavicencio@utec.edu.pe**
Edson Mendiolaza **emendiolaza@utec.edu.pe**
Ian Paul Brossard **ibrossard@utec.edu.pe**
Jose Chavez **jchaveza@utec.edu.pe**
Julio Yarasca **jyarascam@utec.edu.pe**
Percy Quevedo **pquevedo@utec.edu.pe**
Wilder Nina **wnina@utec.edu.pe**
José Fiestas **jfiestas@utec.edu.pe**

Material elaborado por:
Maria Hilda Bermejo

Logro de la sesión:

Al finalizar la sesión, los alumnos se familiarizan con el paradigma de la programación orientada a Objetos.

- Clase – Objeto
- Métodos de acceso (setter y getters)
- Constructores y destructores
- Objetos dinámicos
- Array de objetos

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA

Logro de la sesión:

Al finalizar la sesión, los alumnos se familiarizan con el paradigma de la programación orientada a Objetos.

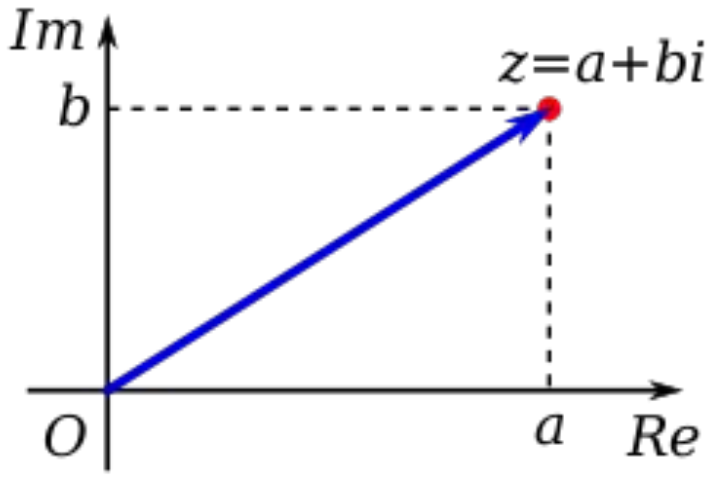
- Clase – Objeto
- Métodos de acceso (setter y getters)
- Constructores y destructores
- Objetos dinámicos
- Arreglo de objetos

5.3

Unidad 5: POO
Objetos dinámicos

Ejercicio 1

- Crear una clase **Complejo** que tenga como atributos los valores **real** e **imaginario** de un número complejo
- Además, un método de clase que realice la **suma** y **diferencia** de dos números complejos
- Almacene los números complejos en un **vector** e **imprima** los números desde el vector

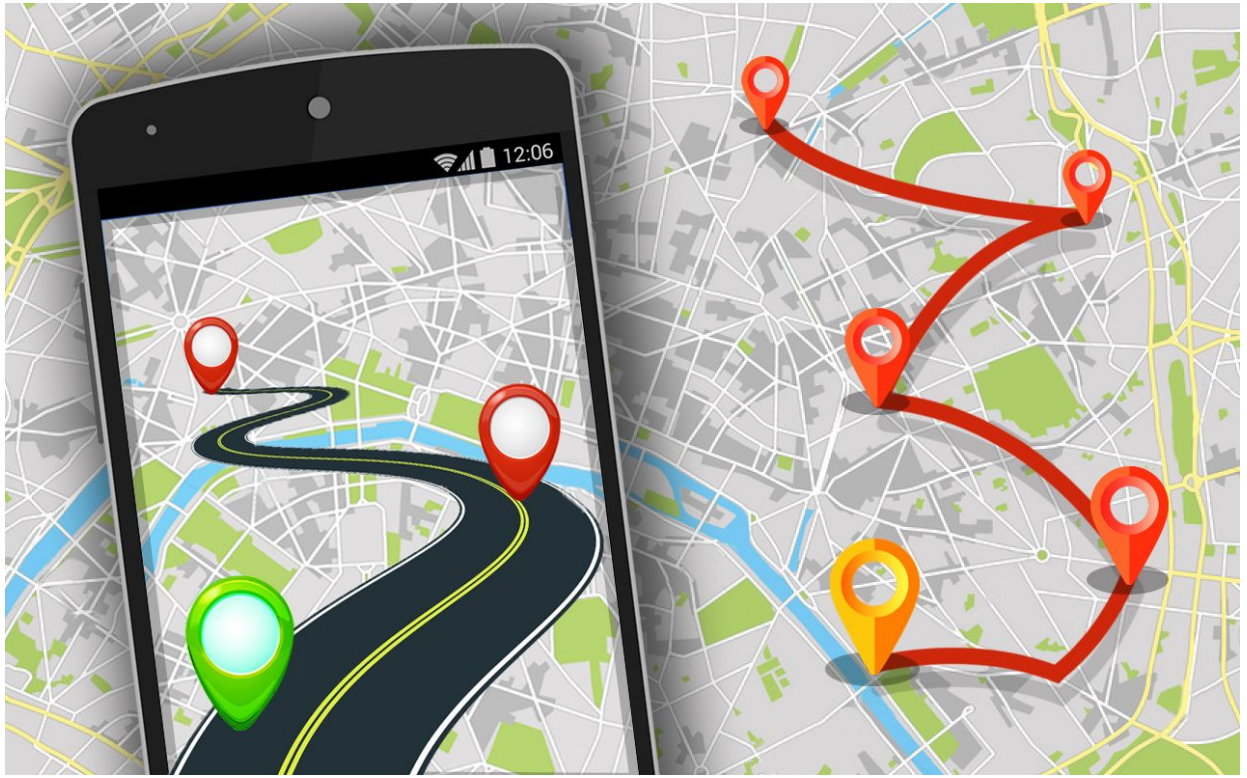


$$(a + ib) + (c + id) = (a + c) + i(b + d)$$

$$(a + ib) - (c + id) = (a - c) + i(b - d)$$

Ejercicio 2

Crear una clase **GPS** que represente la posición (coordenadas) de una persona o vehículo en una ruta.
La ruta almacenará objetos de tipo GPS en un **vector**, que representa la ruta seguida
Utilice la librería ***ctime*** para almacenar la hora actual de la posición en cada punto de la ruta e imprimir la ruta seguida.
Debe convertir la hora a string antes de poder imprimirla



Los siguientes ejercicios utilizan plantillas (se desarrollará este tema en las siguientes clases)

Ejercicio 3

Puntero a objetos

Recordar que en el laboratorio anterior se creó un método genérico para imprimir los datos básicos tanto para la clase CAlumno como para la clase CProfesor: nombre, apellidos y edad.

```
template <typename T>
void mostrarDatosBasicos(T &objeto)
```

- **Ahora se le pide modificar dicha función para recibir el parámetro ya no por referencia sino como puntero.**
- **Además debe crear otra función para solicitar los datos básicos por teclado.**

Ejercicio 3

Main.cpp

```
#include "Funciones.h" //implementar aqui las fun
ciones

int main() {
    cout<<"Alumno:\n";
    CAlumno *a1 = new CAlumno();
    pedirDatosBasicos<CAlumno>(a1);

    cout<<"Profesor:\n";
    CProfesor* p1 = new CProfesor();
    pedirDatosBasicos<CProfesor>(p1);

    cout<<“Mostrar:\n";
    mostrarDatosBasicos<CAlumno>(a1);
    mostrarDatosBasicos<CProfesor>(p1);

    delete a1; delete p1;

    return 0;
}
```

```
Alumno:
--------
Nombre: Jorge
Apellidos: Arriola Carranza
Edad: 19

Profesor:
--------
Nombre: Maria
Apellidos: Palacios Guevara
Edad: 35

Mostrar:
--------
Nombre: Jorge
Apellidos: Arriola Carranza
Edad: 19
--------
Nombre: Maria
Apellidos: Palacios Guevara
Edad: 35
```

Ejercicio 3

Funciones.h

```
#include <iostream>
#include "CAlumno.h"
#include "CProfesor.h"

template <typename T>
void mostrarDatosBasicos(T* objeto)
{
    cout<<"--------\n";
    cout<<"Nombre: "<<objeto->getNombre()<<endl;
    cout<<"Apellidos: "<<objeto->getApellidos()<<endl;
    cout<<"Edad: "<<objeto->getEdad()<<endl;
}

template <typename T>
void pedirDatosBasicos(T* objeto)
{
    cout<<"--------\n";
    texto temps; entero tempe;
    cout<<"Nombre: ";
    getline(cin, temps); objeto->setNombre(temps);
    cout<<"Apellidos: ";
    getline(cin, temps); objeto->setApellidos(temps);
    cout<<"Edad: ";
    cin>>tempe; objeto->setEdad(tempe);
    cin.ignore();
}
```

Ejercicio 4

En el ejercicio 2 de la clase anterior hemos implementado la clase CVector pero que solo soporta arreglos de números enteros. Que pasa si queremos operar con números reales, o strings. ¿Es necesario crear otra clase equivalente para cada tipo de dato?

Pues no, ahora vamos a generalizar nuestra clase CVector para soportar cualquier tipo de dato en el arreglo dinámico. Para ello usaremos el concepto de **templates** en C++.

```
template <class|typename <id>[,...]>
class <identificador_de_plantilla> {
 // Declaración de funciones
 // y datos miembro de la plantilla
};
```

Ejercicio 4

Main.cpp

```cpp
#include "TVector.h"

int main()
{
    //vector sin inicializar
    TVector<double> vector1();

    //vector inicializado con tamaño 10 y elementos inicializado
s con 1
    TVector<double>  vector2(5, 1.0);

    //vector inicializado a partir de otro vector
    TVector<double>  vector3(vector2);//[1.0,1.0,1.0,1.0,1.0]

    //reemplazar el valor de un elemento
    vector3.setElemento(20.0, 0);//[20.0,1.0,1.0,1.0,1.0]
    //insertar elemento en una posicion especifica
    vector3.insertar(25.0, 4);//[20.0,1.0,1.0,1.0,25.0,1.0]
    //agregar elemento al final del vector
    vector3.agregar(10.0);//[20.0,1.0,1.0,1.0,25.0,1.0,10.0]

    //obtener el tamaño actual del vector
    entero n = vector3.getTamanio();//n = 7
    for(int i=0; i<n; i++)
        cout<<vector3.getElemento(i)<<endl;

    return 0;
}
```

TVector.h

```cpp
#include <iostream>
using namespace std;
typedef int entero;

template <typename T>
class TVector {
private:
    entero tamanio;
    T* elementos;
    entero maximo;
    void redimensionar();
public:
    //constructor vacio
    TVector();
    //constructor inicilizador
    TVector(entero _tamanio, T _valor);
    //constructor copia
    TVector(const TVector &otro_vector);
    //destructor
    virtual ~TVector();
    //reemplazar elemento
    void setElemento(T elemento, entero posicion);
    //insertar elemento
    void insertar(T elemento, entero posicion);
    //agregar elemento
    void agregar(T elemento);
    //getters
    entero getTamanio();
    T getElemento(entero posicion);
};
```

¡Nos vemos en la siguiente clase!

UTEC
UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA