

Templates

TC1031 Programación de estructura de datos y
Algoritmos Fundamentales

Ing. Luis Humberto González Guerra
Mtra. Alejandra De Luna Pámanes
aledelunap@tec.mx

Agosto 10, 2021

Templates

Muchas veces escribimos una función o una clase que cumple con cierta funcionalidad, pero es dependiente del tipo de dato. En caso de utilizar otro tipo de dato, se tendría que hacer otra función, lo cual no es eficiente.

- Por ejemplo, consideremos el caso de encontrar el elemento mas grande en un array. Un algoritmo simple consiste en un escaneo secuencial, en el cual se examina en orden cada elemento del array, guardando el elemento mas grande encontrado hasta el momento. Este algoritmo de búsqueda, como muchos otros algoritmos, es independiente del tipo de elementos que contiene el array, ya que pueden ser enteros, decimales, o cual quiere tipo para el cual se pueda definir una comparación.

Templates

Ejemplo

```
int maximo_int(int a, int b)
{
    if (a > b) {
        return a;
    }
    else {
        return b;
    }
}
```

```
double maximo_double(double a, double b)
{
    if (a > b) {
        return a;
    }
    else {
        return b;
    }
}
```

Templates

Ejemplo de función

En C++ podemos escribir algoritmos o estructuras de datos independientes del tipo. Estos algoritmos, también llamados algoritmos genéricos, se conocen como Templates (Plantillas).

```
template <typeName T>
T maximo(T a, T b)
{
    if (a > b) {
        return a;
    }
    else {
        return b;
    }
}
```

Templates

Ejemplo de función

```
maximo<int>(3, 7);
```

```
int maximo_int(int a, int b)
{
    if (a > b) {
        return a;
    }
    else {
        return b;
    }
}
```

```
maximo<double>(3.5, 3.3);
```

```
double maximo_double(double a, double b)
{
    if (a > b) {
        return a;
    }
    else {
        return b;
    }
}
```

Templates

Ejemplo de clase

```
template <class T>
class Value{
    public:
        Value();
        Value(T value);
        itemType getValue() const;
        void setValue(T value);
    private:
        T value;
};
```

Templates

Ejemplo de clase

```
template <class T>
Value<T>::Value(T value) {
    this->value = value;
}
```

```
template <class T>
itemType Value<T>::getValue() const{
    return this->value;
}
```