

Ayudantía 10

Repaso S2

Benjamín Aceituno

Programación Avanzada S09



**Recordemos polimorfismo,
vectores, map, etc...**



Polimorfismo

El polimorfismo en C++ sirve para hacer responder a distintas clases con una misma función, osea, que cada una funcione de forma distinta al mismo mensaje. Se debe ocupar virtual en la clase base para ocupar esta característica.

Virtual

Permite que las funciones compartidas de la clase base puedan ser modificadas y de esta manera se puedan distinguir cuando se usan

Ejemplo:

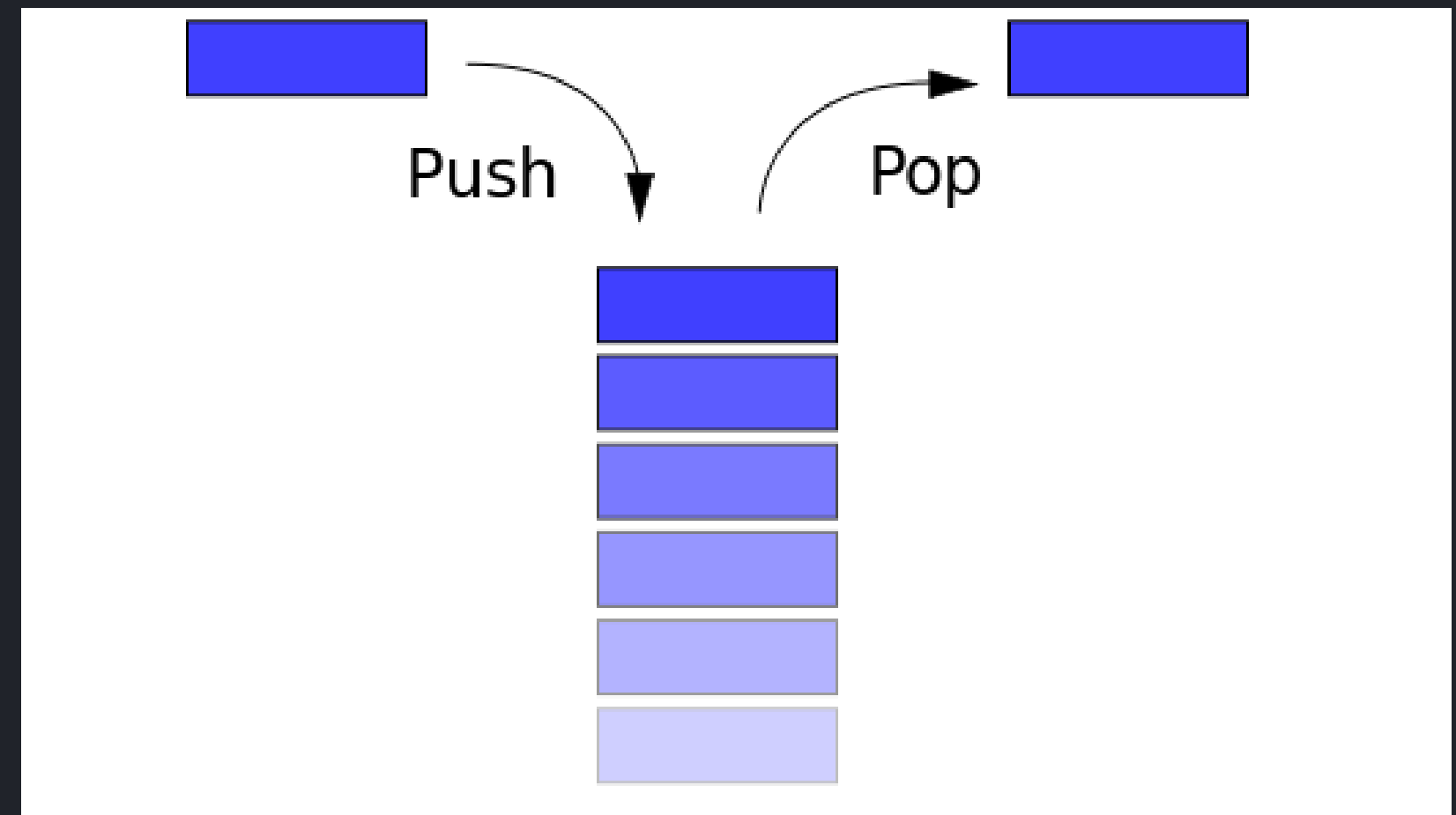
Tenemos la clase base Animal y las clases derivadas Gato y Perro, las tres clases como tal pueden hacer un sonido, usando virtual en la clase Animal podemos hacer que se llame al método virtual de la clase base a través de las clases Perro y Gato.

Vector

Los vectores en C++ funcionan como un arreglo dinámico, es decir, un arreglo pero sin un tamaño definido. Funciona en base a la metodología LIFO, y su tamaño va variando dependiendo de los elementos que añadimos o eliminamos.

LIFO?..

LIFO quiere decir Last In First Out, es decir, último entrar, primero en salir. Intenten verlo como si fuera almacenar una pila de cajas, nosotros ponemos cajas una encima de otra, y si sacamos una caja será la última que hemos puesto.



Funciones

El vector en C++ se debe incluir de esta manera en el código

```
#include <vector>
```

Se declara de esta forma

```
vector<tipoDeDato>  
nombre;
```

```
vector<int> numeros;  
vector<string> palabras;  
vector<Objeto*> objetos;
```

`push_back(item)`: Inserta un ítem

`pop_back()`: Elimina un ítem.

`clear()`: Elimina todo.

`size()`: Retorna el número de elementos del vector.

`at(i)`: Retorna el valor del vector en una posición dada.

`empty()` : Retorna true si el vector está vacío.

Map

Es una estructura de dato que almacena pares Llave valor. Los elementos en un map están ordenados automáticamente según las llaves, y cada llave es única, lo que significa que no puede haber duplicados.

Declaración: `map<llave, valor> nombre;`
Se debe incluir con `#include <map>`

```
#include <map>
using namespace std;

int main() {
    map<string, int> edad;

    edad["Vicente"] = 25;
    edad["Diego"] = 30;
    edad["Javier"] = 20;
}
```

```
map<string, float> notas;
```

Llave		Valor
21.000.000-0	→	6.3
21.000.001-0	→	2.4
21.000.002-0	→	1.0
21.000.003-0	→	6.0
21.000.004-0	→	4.3
21.000.005-0	→	7.0

Funciones

Para agregar un valor al mapa: `map[llave] = valor;`

`begin()`: Retorna el iterador de la primera posición.

`end()`: Retorna el iterador después de la última posición.

`empty()`: Devuelve true si el map está vacío.

`size()`: Devuelve el tamaño del map.

`swap(map)`: Intercambia valores de un map a otro.

`count(llave)`: Retorna true si la llave existe en el map, false en caso contrario.

`find(llave)`: Retorna un iterador al elemento con la clave especificada

Ejercicio 1

Una empresa se dedica a la venta de diversas cosas, y tiene tiendas específicas de cada eso, de las cuales destacaremos la venta de ropa, se tienen distintas sucursales de ropa. Se pide:

Crear clase Empresa

Tiene el nombre del dueño, rut de la empresa y saldo total (las ganancias producto de sus ventas). Debe el método `info()` que imprime los atributos. Además el método `acumularSaldo(ingreso)`, que recibe un monto y lo acumula al saldo total. Se debe crear constructor con los datos de la empresa y el saldo inicia en 0.

Crear clase TiendaRopa: Hereda de Empresa

Contiene un vector que simula los fardos de ropa que se tienen. En el vector se guarda el peso (en kilos) de cada fardo. Cree el método `exportarFardo()` que elimina el último fardo que se tiene y lo retorna. Se debe crear el método `verificarFardos()` que retorna `true` si quedan fardos, `false` en caso contrario. Además el método `ver()` que muestra todos los fardos que hay con su peso.

Crear clase Sucursal1: Hereda de TiendaRopa

Cree el método `venta(cantidad)` que recibe una cantidad de fardos a vender (debe usar método `exportarFardo` y `verificarFardos`) y los vende según disponibilidad. El kilo está a 1000 pesos, esto se debe sumar a las ganancias de la empresa con `acumularSaldo`.

Ejercicio 2

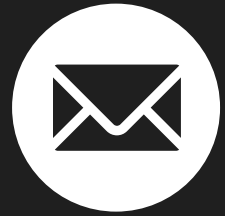
Desde la empresa Cerbel se le pide que realice un programa para contar votos por comuna y obtener sus porcentajes respecto a una ciudad (Una ciudad es compoene por comunas) Para esto, tiene que crear la clase Elecciones que contiene un map con el nombre de la comuna y un stack de los votos de la comuna.

Debe crear los métodos agregarVoto que agrega una cantidad de votos respecto a una comuna, mostrarVotos que muestra el nombre de la comuna y los votos que se ingresaron, contarVotos, que recibe la comuna y cuentas los votos que hay, también debe indicar si no hay votos registrados. contarVotosPorCiudad que cuenta todos los votos que hay en la ciudad y obtiene los porcentajes por comuna.

Contacto



+569 86031881



benjamin.aceituno@mail.udp.cl



benja.mp4



[https://github.com/benjamp4/
prog.av-2024-2](https://github.com/benjamp4/prog.av-2024-2)

