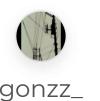
#### WUOLAH



www.wuolah.com/student/gonzz\_



# **Lineacpp.pdf**Examen ordinaria 2018 RESUELTO

- i 1º Metodología de la Programación
- © Grado en Ingeniería Informática
- Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
  Universidad de Granada

# **WUOLAH** + #QuédateEnCasa

#### #KeepCalm #EstudiaUnPoquito

**Enhorabuena**, por ponerte a estudiar te **regalamos un cartel** incluído entre estos apuntes para estos días.

```
#include "Linea.h"
Linea:: Linea(int num_paradas){
       assert(num_paradas>= 0);
       if (num_paradas== 0){
              this->num_paradas= 0;
              paradas= 0;
       } else {
              ReservaMemoria(num_paradas);
       }
}
Linea:: ~Linea(){
       LiberaMemoria();
}
Linea:: Linea(const Linea &linea){
       paradas= 0;
       num_paradas= 0;
       *this = linea;
}
int Linea:: GetNumParadas()const{
       return num_paradas;
}
bool Linea:: EstaVacia() const{
       return paradas== 0;
}
bool Linea:: IndiceValido(int i) const{
       return i>= 0 && i < num_paradas;
}
void Linea:: LiberaMemoria(){
       if (!EstaVacia()){
              delete [] paradas;
              paradas= 0;
       }
}
void Linea:: ReservaMemoria(int tam){
       assert (tam > 0);
       paradas= new InfoParada[tam];
       num_paradas= tam;
}
```



Reservados todos los derechos. No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.



### Gana dinerito extra.

## Recomienda a tus negocios favoritos que se anuncien en Wuolah y llévate 50€.

Te daremos un código promocional para que puedan anunciarse desde 99€.

1 Ve a tu negocio favorito • 2 Dales tu código de promo

• 3 Diles que nos llamen o nos escriban.



```
void Linea:: CopiaParadas(const Linea & linea){
       for(int i= 1; i <= linea.num_paradas; i++){</pre>
              (*this)[i]= linea[i];
}
Linea& Linea:: operator=(const Linea & linea){
       if(&linea!=this){
              //No compruebo paradas != 0 ya que lo compruebo dentro
              LiberaMemoria();
              ReservaMemoria(linea.num_paradas);
              CopiaParadas(linea);
       return * this;
}
InfoParada& Linea:: operator[](int i){
       IndiceValido(i);
       return paradas[i-1];
}
InfoParada Linea:: operator[](int i)const{
       IndiceValido(i);
       return paradas[i-1];
}
Linea & Linea:: operator+=(const InfoParada & parada){
       Linea nueva(num_paradas+1);
       nueva.CopiaParadas(*this);
       nueva[num_paradas+1]= parada;
       LiberaMemoria();
       *this= nueva:
       return *this;
}
ostream& operator<<(ostream & flujo, const Linea & linea){
       flujo<< "Número de paradas: " << linea.GetNumParadas() << endl;
       for(int i= 1; i<=linea.GetNumParadas(); i++)</pre>
               flujo<<"\tParada " << i << ": " << linea[i] << endl;
       return flujo;
}
int Linea:: GetNumParadasActivas()const{
       int paradas_activas= 0;
       for(int i= 1; i <= num_paradas; i++){
              if((*this)[i].EstaActiva())
```





```
paradas_activas++;
}
return paradas_activas;
}
```

