Programación Avanzada I
Tarea 15

Andrés César Magaña Martínez

Problema 3.

Los constructores virtuales existen para hacer algo que C++ no soporta directamente. Los constructores no se pueden declarar virtuales, los metodos create() y clone() sirven para definir un constructor por default y un constructor por copia respectivamente.

```
01   class Shape {
02   public:
03      virtual ~Shape() { }                  // Destructor virtual
04      ...
05      virtual Shape* clone()  const = 0;    // Usa el constructor por copia
06      virtual Shape* create() const = 0;    // Usa el constructor por default
07      };
08
09   class Circle : public Shape {
10   public:
11      Circle* clone()  const;
12      Circle* create() const;
13      ...
14   };
15
16   Circle* Circle::clone()  const {
17   return new Circle(*this);
18   }
19
20   Circle* Circle::create() const {
21   return new Circle();
22   }
```

En la función clone(), new Circle(*this) llama al constructor por copia. En la función create(), new Circle() llama al constructor por default.

Después se pueden utilizar estos métodos como si fueran constructores:

```
01   void userCode(Shape& s)
02      {
03      Shape* s2 = s.clone();
04      Shape* s3 = s.create();
05      ...
06      delete s2;    // Aquí se llama al destructor virtual
07      delete s3;
08      }
```