



Pràctica 0

Fitxa 5: Exercici de C++

GiVD - curs 2022-2023

- **Exercici 1:** Amplia el subprojecte de **Figures** amb les següents funcionalitats:
- Arega al projecte una classe **FigureContainer**. Aquesta ha de tenir:
 - Un `std::vector` privat de punters a objectes **Figure**. Quina diferència hi ha entre usar el tipus vector o un array de C++? Es possible en aquest cas utilitzar un array de C++?
 - Un mètode públic **void addFigure (Figure*)** que afegirà al vector la figura que se li passi per paràmetre.
 - Un mètode públic **double getPerimeters ()** que retornarà la suma dels perímetres de les figures contingudes en el vector. Per recórrer el vector pots utilitzar un recorregut pel nombre d'elements, un iterador o la sentència `foreach` (si uses un compilador de C++11).
- En el main crea un objecte **FigureContainer** i arega els **Square** i **Circle** creats prèviament, obté la suma dels seus perímetres.
- Comprova que es criden a tots els constructors i destructors de les classes. Es destrueix tota la memòria dinàmica allocatada? Utilitza els cast dinàmics per a poder esborrar cadascuna de les figures. Per exemple, per esborrar una figura quadrat:

```
if (dynamic_cast<Square*>(figure[i]))  
    delete (Square *) (figure[i]);
```

Exercici 2:

- Arega un mètode **void setRadius (double)** a la classe **Circle**, que canviarà el valor del radi de l'objecte **Circle**
- En el main, després d'obtenir la suma dels perímetres de les figures en **FigureContainer**, canvia el radi de l'objecte **Circle** ja creat i torna a obtenir la suma dels perímetres afegits al contenidor de figures. Observa el resultat executant-lo en l'entorn de QtCreator.
- Ressegueix amb el debugger els diferents passos de l'execució posant un breakpoint al mètode `setRadius`.