Pau Bernabé - NIUB: 20081736
Marc Cerrillo - NIUB: 20081552

Pràctica 1 - Estructura de Dades

ÍNDEX

Exercici 1:

En aquest exercici hem de crear un programa *main.cpp* que demana el nom de l'usuari i depenent de l'opció dóna la benvinguda o sortida del programa.

Primer cridem a la llibreria *iostream* i el *namespace*.

Dins el *main* declarem les variables i creem un array per escollir les opcions. Després posem un *cout* que demani el nom i un *cin* que llegeixi aquest.

Un cop entra el nom el programa entra en un *do{...}while(option != 0)* per escollir la opció desitjada. Posem un *cin* per llegir la opció i creem un *if* i un *else if* per retornar la resposta corresponent. Fins que no entri la opció 1 no surt del bucle.

Exercici 2:

En aquest exercici hem de fer unes modificacions i ampliacions de l'exercici anterior.

Creem un mètode que li diem *seeMenu* amb un vector de *strings* de 3 espais per posar les diferents opcions i un bucle *for* perquè surti per pantalla les 3 opcions disponibles.

Dins el *do while* del mètode *main* eliminem l'array de l'exercici anterior i cridem al mètode *seeMenu*. Un cop entri la opció desitjada implementem un *switch* que segons la opció escollida retorni la resposta corresponent.

Exercici 3:

En aquest exercici hem de crear una classe *Circle* amb un mètode *getArea* que et demana per pantalla el radi del teu cercle i et retorna l'àrea d'aquest. Dins la classe *circle.cpp* hem d'enviar una excepció si el radi és 0 o negatiu.

Primer creem la classe *Cercle* amb els atributs privats i els mètodes públics. Implementem el constructor *Cercle* i els mètodes *getArea()*, *setRadi()* i *getRadi()*.

Dins el *switch* fem un *try….catch* si el radi és 0 o negatiu per controlar l'error. Si el número introduït és adequat cridem als mètodes de la classe *cercle* perquè llegeixi el radi i ens doni l'àrea. Un cop fet sumem 1 al comptador dels cercles i torna al menú inicial.

Exercici 4:

En aquest exercici hem de canviar la classe *Circle* per una classe *Ellipse* que ens demani el/s radi/s d'un cercle o un el·lipse i ens retorni la seva àrea.

Dins la classe *Ellipse* declarem els mètodes del cercle i de l'el·lipse amb i sense paràmetres. Dins el *switch* segons la lletra introduïda entra a un *if* o a l'altre. Comprovem que no hi hagi errors amb els radis i llavors cridem als mètodes per calcular l'àrea del cercle o de l'el·lipse segons la lletra que hagi introduït.

Exercici 5:

En aquest exercici ens demana llegir un fitxer i que ens dibuixi per pantalla l'àrea de la figura.

Primer cridem a la llibreria *fstream* per treballar amb fitxers. Amb el *ifstream* obrim el fitxer que hem creat amb les dades que ens donaven. Si el fitxer està obert el *while(!file.eof())* el llegira fins el final. Llegim el tipus de figura que és (C si és cercle i E si és el·lipse), llegim el radi i cridem als mètodes corresponents per calcular l'àrea. Un cop hem llegit el fitxer el tanquem (file.close()) i tornem al menú inicial.

Exercici 6:

En aquest exercici ens demana crear diverses classes per treballar amb herència.

Implementem constructors sense paràmetres i cada vegada que es crei un cercle o un el·lipse ens digui per pantalla "Hola sóc el constructor del cercle/el·lipse". Veiem que quan s'executa el constructor de l'el·lipse ens diu que és el constructor d'aquest i amb el cercle surt el constructor d'el·lipse i de cercle per la relació d'herència.

El mètode *getArea()* virtual ens permet manipular el mètode de la classe mare/base a les seves classes filles/derivades.