```
1   /*
2    * Proyecto: PunterosEnHerencia
3    * Archivo:  Base.h
4    * Autor:    J.  Miguel Guanira E. //miguel.guanira.
5    *
6    * Created on 28 de mayo de 2024, 08:41 AM
7    */
8
9
10  #ifndef BASE_H
11  #define BASE_H
12
13  class Base {
14  private:
15      int b;
16  public:
17      void SetB(int b);
18      int GetB() const;
19      virtual void mostrar();
20      virtual void leerdatos();
21  };
22
23  #endif /* BASE_H */
24
25  /*
26   * Proyecto: PunterosEnHerencia
27   * Archivo:  Base.cpp
28   * Autor:    J. Miguel Guanira E. //miguel.guanira.
29   *
30   * Created on 28 de mayo de 2024, 08:41 AM
31   */
32
33  #include <iostream>
34  #include <iomanip>
35  using namespace std;
36
37  #include "Base.h"
38
39
40  void Base::SetB(int b) {
41      this->b = b;
42  }
43
44  int Base::GetB() const {
45      return b;
46  }
47
48  void Base::mostrar() {
49      cout<<"Estamos en mostrar de la clase base"<<endl;
50      cout<<"B= "<<b<<endl;
51  }
52
53  void Base::leerdatos(){
54      cin>>b;
55  }
56
57  /*
58   * Proyecto: PunterosEnHerencia
59   * Archivo:  Derivada.h
60   * Autor:    J.  Miguel Guanira E. //miguel.guanira.
61   *
62   * Created on 28 de mayo de 2024, 08:47 AM
63   */
64
65
66  #ifndef DERIVADA_H
```

```
67  #define DERIVADA_H
68  #include "Base.h"
69  class Derivada : public Base{
70  private:
71      int d;
72  public:
73      void SetD(int d);
74      int GetD() const;
75      void mostrar();
76      void leerdatos();
77  };
78
79  #endif /* DERIVADA_H */
80
81  /*
82   * Proyecto: PunterosEnHerencia
83   * Archivo:  Derivada.cpp
84   * Autor:    J. Miguel Guanira E. //miguel.guanira.
85   *
86   * Created on 28 de mayo de 2024, 08:47 AM
87   */
88
89  #include <iostream>
90  #include <iomanip>
91  using namespace std;
92
93  #include "Derivada.h"
94
95  void Derivada::SetD(int d) {
96      this->d = d;
97  }
98
99  int Derivada::GetD() const {
100     return d;
101 }
102
103 void Derivada::mostrar() {
104     cout<<"Estamos en mostrar de la clase derivada"<<endl;
105     cout<<"B= "<<GetB()<<endl;
106     cout<<"D= "<<d<<endl<<endl;
107 }
108
109 void Derivada::leerdatos() {
110     Base::leerdatos();
111     cin>>d;
112 }
113
114 /*
115  * Proyecto: PunterosEnHerencia
116  * Archivo:  Derivada2.h
117  * Autor:    J.  Miguel Guanira E. //miguel.guanira.
118  *
119  * Created on 28 de mayo de 2024, 09:04 AM
120  */
121
122
123 #ifndef DERIVADA2_H
124 #define DERIVADA2_H
125 #include "Base.h"
126
127 class Derivada2 : public Base{
128 private:
129     int d2;
130 public:
131     void SetD2(int d2);
132     int GetD2() const;
```

```
133         void mostrar();
134     };
135
136     #endif /* DERIVADA2_H */
137
138     /*
139      * Proyecto: PunterosEnHerencia
140      * Archivo:  Derivada2.cpp
141      * Autor:    J. Miguel Guanira E. //miguel.guanira.
142      *
143      * Created on 28 de mayo de 2024, 09:04 AM
144      */
145
146     #include <iostream>
147     #include <iomanip>
148     using namespace std;
149
150     #include "Derivada2.h"
151
152
153     void Derivada2::SetD2(int d2) {
154         this->d2 = d2;
155     }
156
157     int Derivada2::GetD2() const {
158         return d2;
159     }
160
161     void Derivada2::mostrar() {
162         cout<<"Estamos en mostrar de la clase derivada 2"<<endl;
163         cout<<"B= "<<GetB()<<endl;
164         cout<<"D2= "<<d2<<endl<<endl;
165     }
166
167     /*
168      * Proyecto: PunterosEnHerencia
169      * Archivo:  main.cpp
170      * Autor:    J. Miguel Guanira E.//miguel.guanira.
171      *
172      * Created on 28 de mayo de 2024, 08:41 AM
173      */
174
175     #include <iostream>
176     #include <iomanip>
177     using namespace std;
178     #include "Derivada.h"
179     #include "Derivada2.h"
180     void f(class Base*pt){
181         pt->mostrar();
182     }
183     int main(int argc, char** argv) {
184     //    class Base base;
185         class Derivada derivada11, derivada12;
186         class Derivada2 derivada21, derivada22;
187     //    class Base *pt;
188     //    base.SetB(555);
189         derivada11.SetB(1111);
190         derivada11.SetD(1333);
191         derivada12.SetB(2111);
192         derivada12.SetD(2333);
193         derivada21.SetB(1101);
194         derivada21.SetD2(1505);
195         derivada22.SetB(2101);
196         derivada22.SetD2(2505);
197
198     //    base.mostrar();
```

```
199   //
200   //    derivada.mostrar();
201   //
202   //    derivada2.mostrar();
203   //
204   //    pt = &derivada;
205   //    pt->mostrar();
206   
207   //    pt = &derivada2;
208   //    pt->mostrar();
209   
210   //    f(&derivada2);
211   //    f(&derivada);
212         class Base *arr[5];
213         arr[0] = &derivada21;
214         arr[1] = &derivada11;
215         arr[2] = &derivada12;
216         arr[3] = &derivada22;
217         for(int i=0; i<4; i++)
218             arr[i]->mostrar();
219   
220         return 0;
221   }
222   
```