

```
1  /*
2  * Proyecto: PunterosEnHerencia
3  * Archivo: Base.h
4  * Autor: J. Miguel Guanira E. //miguel.guanira.
5  *
6  * Created on 29 de octubre de 2024, 08:40 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     // void imprime();
20     virtual void imprime();
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 29 de octubre de 2024, 08:40 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::imprime() {
49     cout<<"Estoy en la clase Base"<<endl;
50     cout<<"B = "<<b<<endl;
51 }
52
53 /*
54 * Proyecto: PunterosEnHerencia
55 * Archivo: DERivadal.h
56 * Autor: J. Miguel Guanira E. //miguel.guanira.
57 *
58 * Created on 29 de octubre de 2024, 08:42 AM
59 */
60
61
62 #ifndef DERIVADA1_H
63 #define DERIVADA1_H
64 #include "Base.h"
65
66 class DERivadal : public Base{
```

```
67     private:
68         int d1;
69     public:
70         void SetD1(int d1);
71         int GetD1() const;
72         void imprime();
73 };
74
75 #endif /* DERIVADA1_H */
76
77 /*
78  * Proyecto: PunterosEnHerencia
79  * Archivo:  DErivada1.cpp
80  * Autor:    J. Miguel Guanira E. //miguel.guanira.
81  *
82  * Created on 29 de octubre de 2024, 08:42 AM
83  */
84
85 #include <iostream>
86 #include <iomanip>
87 using namespace std;
88
89 #include "DErivada1.h"
90
91
92 void DErivada1::SetD1(int d1) {
93     this->d1 = d1;
94 }
95
96 int DErivada1::GetD1() const {
97     return d1;
98 }
99
100 void DErivada1::imprime() {
101     cout<<"Estoy en la clase Derivada 1"<<endl;
102     // cout<<"B = "<<GetB()<<endl;
103     Base::imprime();
104     cout<<"D1 = "<<d1<<endl;
105 }
106
107 /*
108  * Proyecto: PunterosEnHerencia
109  * Archivo:  Derivada2.h
110  * Autor:    J. Miguel Guanira E. //miguel.guanira.
111  *
112  * Created on 29 de octubre de 2024, 08:42 AM
113  */
114
115
116 #ifndef DERIVADA2_H
117 #define DERIVADA2_H
118 #include "Base.h"
119
120 class Derivada2 : public Base {
121 private:
122     int d2;
123 public:
124     void SetD2(int d2);
125     int GetD2() const;
126     void imprime();
127 };
128
129 #endif /* DERIVADA2_H */
130
131 /*
132  * Proyecto: PunterosEnHerencia
```

```
133  * Archivo:   Derivada2.cpp
134  * Autor:     J. Miguel Guanira E. //miguel.guanira.
135  *
136  * Created on 29 de octubre de 2024, 08:42 AM
137  */
138
139  #include <iostream>
140  #include <iomanip>
141  using namespace std;
142
143  #include "Derivada2.h"
144
145  void Derivada2::SetD2(int d2) {
146      this->d2 = d2;
147  }
148
149  int Derivada2::GetD2() const {
150      return d2;
151  }
152
153  void Derivada2::imprime() {
154      cout<<"Estoy en la clase Derivada 2"<<endl;
155      // cout<<"B = "<<GetB()<<endl;
156      Base::imprime();
157      cout<<"D2 = "<<d2<<endl;
158  }
159
160  /*
161  * Proyecto: PunterosEnHerencia
162  * Archivo:   main.cpp
163  * Autor:     J. Miguel Guanira E.//miguel.guanira.
164  *
165  * Created on 29 de octubre de 2024, 08:38 AM
166  */
167
168  #include <iostream>
169  #include <iomanip>
170  #include <random>
171  using namespace std;
172  #include "Base.h"
173  #include "DERivada1.h"|
174  #include "Derivada2.h"
175
176  void f(class Base *pt){
177      pt->imprime();
178  }
179
180
181  int main(int argc, char** argv) {
182      class Base base, base0;
183      class DERivada1 der1,der11;
184      class Derivada2 der2,der22;
185      class Base *pt[10];
186
187      base.SetB(5);
188      der1.SetB(15);
189      der1.SetD1(115);
190      der2.SetB(25);
191      der2.SetD2(225);
192      //// base.imprime();
193      //// der1.imprime();
194      //// der2.imprime();
195      //
196      //// pt = &base;
197      // pt = &der1;
198      // pt->imprime();
```

```
199 //
200 //     pt = &der2;
201 //     pt->imprime();
202 //
203     base0.SetB(7);
204     der11.SetB(17);
205     der11.SetD1(117);
206     der22.SetB(27);
207     der22.SetD2(227);
208
209     pt[0]= &der11;
210     pt[1]= &base0;
211     pt[2]= &der2;
212     pt[3]= &der22;
213     pt[4]= &der1;
214     pt[5]= &base;
215
216     for(int i=0; i<6; i++){
217         pt[i]->imprime();
218     }
219
220     f(&der22);
221
222     return 0;
223 }
224
```