

```
1 // field2.cpp
2 //
3 // Prikklad pretizeneho prirazovaciho operatoru
4
5 // Vzhledem k rozdilum v chovani prekladacu
6 // zde nekontrolujeme, zda se alokace podarila, stejne jako
7 // v predchozich verzich
8
9 #ifndef _WIN32
10     #include <tchar.h>
11     #include <conio.h>
12 #elif (defined __linux__) || (defined _AIX)
13     typedef char _TCHAR;
14     #define _tmain main
15 #endif
16
17 #include <iostream>
18 using namespace std;
19
20 class pole {
21     int* p;
22 public:
23     enum { delka = 10 };
24     explicit pole(int d = 0);
25     pole(pole&);
26     int& operator[](int);
27     ~pole() { delete p; }
28     pole& operator=(pole& p);
29 };
30
31 pole& pole::operator=(pole& P) // Pretizeny operator =
32 {
33     for (int i = 0; i < delka; i++)
34         p[i] = P.p[i];
35     return *this; // Zde vraci odkaz na levy operand
36 }
37
38 pole::pole(int d)
39 {
40     p = new int[delka];
41     for (int i = 0; i < delka; i++)
42         p[i] = d;
43 }
44
45 pole::pole(pole& P)
46 {
47     p = new int[delka];
48     for (int i = 0; i < delka; i++)
49         p[i] = P.p[i];
50 }
51
52 int& pole::operator [](int index)
53 {
```

```
54     return p[index];
55 }
56
57 void my_getch();
58
59 int main()
60 {
61     pole P(0);
62     for (int i = 0; i < pole::delka; i++)
63         P[i] = 10;
64     pole Q(0);
65     Q = P;    // Pouzijeme pretizeny prirazovaci operator
66             // .....
67     cout << "\n  There is no output in this example.\n";
68     my_getch();
69     return 0;
70 }
71 //-----
72 void my_getch()
73 {
74 #ifdef _WIN32
75     _getch();
76 #else
77     cout << endl;
78 #endif
79 }
80 //-----
81
```