

```
1 // arrow.cpp
2 //
3 // Příklad pretizeneho operatoru ->
4 // Trida p_datum zapouzduje ukazatel na strukturu datum
5 // a zprostredkovava pristup k ni prostrednictvim pretizeneho
6 // operatoru ->
7 // Destruktor teto tridy pri zaniku instance automaticky
8 // uvolni strukturu, na kterou instance ukazuje
9
10 #ifdef _WIN32
11     #include <tchar.h>
12     #include <conio.h>
13 #elif (defined __linux__) || (defined _AIX)
14     typedef char _TCHAR;
15     #define _tmain main
16 #endif
17
18 #include <iostream>
19 using namespace std;
20
21 struct datum {
22     int den, mesic, rok;
23     datum(int d, int m, int r) : den(d), mesic(m), rok(r) {}
24 };
25
26 class p_datum {
27     datum* p;
28 public:
29     p_datum(datum* pd) : p(pd) {}
30     ~p_datum() { delete p; }
31     datum* operator->() { return p; }
32 };
33
34 void my_getch();
35
36 int main()
37 {
38     p_datum d = new datum(11, 12, 2000);
39     // Nemusime se starat o zruseni
40     cout << endl << d->den << " " << d->mesic << " " << d->rok << endl;
41     my_getch();
42     return 0;
43 }
44 //-----
45 void my_getch()
46 {
47 #ifdef _WIN32
48     _getch();
49 #else
50     cout << endl;
51 #endif
52 }
53 //-----
```