

```
1 // casting.cpp
2 //
3 // Operator pretypovani
4
5 // Trida string reprezentujici znakovy retezec obsahuje pole znaku
6 // Konstruktor do tohoto pole ulozi retezec
7 // Operator pretypovani na const char* vrati ukazatel na toto pole
8 // s je instance tridy my_string
9 // V prikazu strcpy(ccc, s) se s automaticky konvertuje na const char*
10
11 #ifdef _WIN32
12     #pragma warning(disable:4996)
13     #include <tchar.h>
14     #include <conio.h>
15 #elif (defined __linux__) || (defined _AIX)
16     typedef char _TCHAR;
17     #define _tmain main
18 #endif
19
20 #include <stdio.h>
21 #include <string.h>
22 #include <iostream>
23 using namespace std;
24
25 class my_string {
26 public:
27     enum { delka = 10 };
28     my_string(const char* = 0);
29     operator const char* ();
30 private:
31     char c[delka + 1]; // Navic 1 znak pro koncovou 0
32 };
33
34 my_string::my_string(const char* C)
35 {
36     for (int i = 0; i < delka + 1; i++) c[i] = 0;
37     if (C) {
38         int n = strlen(C);
39         strncpy(c, C, (n > delka ? delka : n));
40     }
41 }
42
43 my_string::operator const char* ()
44 {
45     return c;
46 }
47
48 void my_getch();
49
50 int main() {
51     my_string s("Ahoj lidi tam");
52     cout << endl << s << endl; // Vypise "Ahoj lidi "
53     // printf("%s\n",s); // Vypise "Ahoj lidi "
```

```
54 char ccc[20];
55 strcpy(ccc, s); // Zde se automaticky pouzije nas operator pretypovani
56 cout << ccc << endl; // Vypise "Ahoj lidi "
57 // char *c = s; // CHYBA - nelze priradit ukazatel na konstantu ukazateli
58 // na nekonstantni objekt
59 // strcpy(s, ccc); // CHYBA - s nelze konvertovat na char*
60 char d = s[1];
61 // s[1] = 'u'; // CHYBA - chceme menit konstantni objekt
62 my_getch();
63 return 0;
64 }
65 //-----
66 void my_getch()
67 {
68 #ifdef _WIN32
69 _getch();
70 #else
71 cout << endl;
72 #endif
73 }
74 //-----
75
```