

```
1: #include <iostream>
2: #include <string>
3: #include <cassert>      // assert()
4: #include <cstdlib>      // rand()
5: #include <ctime>        // time()
6: using namespace std;
7:
8:
9: /** Einlesen einer ganze Zahl solange, bis diese aus dem Intervall [a
,b] ist.
10: *   @param[in]  a   Intervallanfang
11: *   @param[in]  b   Intervallende
12: *   @return     ganze Zahl aus [a,b]
13: *   @warning    Keine Absicherung bzgl. Fehleingaben
14: */
15: int eingabe(const int a, const int b);           // Prototype
16:
17: int eingabe(const int a, const int b)
18: {
19:     assert(a<=b);    // Bricht ab, falls Voraussetzung a<=b nicht erfu
ellt ist
20:     int k(a-1);
21:     do
22:     {
23:         cout << " Gib Zahl aus [" << a << "," << b << "]" ein : ";
24:         cin >> k;
25:     }
26:     while ( !( a<=k && k<=b ) );
27:
28:     return k;
29: }
30:
31: /** Erraten einer gegebenen Geheimzahl @p iSecret aus dem Intervall [
@p a, @p b].
32: *   @param[in]  iSecret Geheimzahl
33: *   @param[in]  a       Intervallanfang
34: *   @param[in]  b       Intervallende
35: *   @return     Anzahl der Versuche
36: *   @warning    Keine Absicherung bzgl. Fehleingaben
37: */
38: int geheimZahl(const int iSecret, const int a, const int b);
// Prototype
39:
40: int geheimZahl(const int iSecret, const int a, const int b)
41: {
42:     int count(0);
43:     int iGuess(a-1);
44:     do
45:     {
46:
47:         cout << " Zahl aus [" << a << "," << b << "]" << endl;
48:         iGuess = eingabe(a,b); // Garantiert [a,b]
49:         ++count;
50:         string ss;
51:         if (iGuess == iSecret)
52:         {
53:             ss = "richtig";
```

```
54:         }
55:         else if (iGuess < iSecret)
56:         {
57:             ss = "zu klein";
58:         }
59:         else
60:         {
61:             ss = "zu gross";
62:         }
63:         cout << iGuess << " ist " << ss << endl;
64:     }
65:     while (iGuess != iSecret);
66:
67:     return count;
68: }
69:
70: int main()
71: {
72:     cout << "Hello world!" << endl;
73:     const int anf(4), ende(14);           // Kann auch via Tastatur ein
gegeben werden
74:
75:     srand ( time(NULL) );                 // Zufallszahlengenerator sta
rted "zufaellig"
76:     const int is = rand() % (ende-anf+1)+anf; // ganze Zufallszahl a
us [anf, ende]
77:
78:     const int cc = geheimZahl(is, anf, ende);
79:     cout << cc << " Versuche gebraucht!" << endl;
80:
81:     return 0;
82: }
```