

NAKLJUČNA ŠTEVILA

1

PSEVDO NAKLJUČNA ŠTEVILA
GENERATOR NAKLJUČNIH ŠTEVIL
SEME GENERATORJA NAKLJUČNIH ŠTEVIL

Naključna števila

2

- v računalniku ne obstajajo
- psevdo-naključna števila
 - so deterministična
 - so napovedljiva, poznamo pravilo, po katerem so določena
 - zdijo se naključna
- generator naključnih števil
 - funkcija, ki vrne naključno število
 - tvori zaporedje števil
- seme generatorja naključnih števil
 - kje v zaporedju začnemo
 - običajno določeno glede na sistemsko uro

Naključna števila

3

- določena s pomočjo semena
- zakaj sistemska ura?
 - je nepredvidljivo (točen čas, ob katerem zaženemo program)
 - verjetnost, da bo program izveden dvakrat ob istem času je zelo majhna
- generiranje naključnih števil poteka v dveh fazah
 1. določi seme
 2. generiraj naključna števila

Seme naključnih števil

4

- dostop do sistemske ure
 - vključiti moramo knjižnico ctime
 - ✦ #include <ctime>
- določanje semena in generator naključnih števil
 - vključiti moramo knjižnico cstdlib
 - ✦ #include <cstdlib>
- s funkcijo srand(x) določimo seme x
 - x je seme
 - odvisno od časa – uporabimo funkcijo time(o)

```
srand(time(0));
```

običajno določimo **samo enkrat** v programu

Določanje naključnega števila

5

- naključno število dobimo s klicem funkcije rand()
 - `int x = rand();`
 - ✦ x je celo število
 - ✦ $0 \leq x < \text{RANDMAX}$
 - ✦ RANDMAX je konstanta definirana v knjižnici cstdlib
 - ✦ dobimo lahko zelo velika cela števila
- naključno število med *min* in *max*
 - *min* ... najmanjše število, ki ga želimo
 - *max* ... največje število, ki ga želimo

```
x=rand()%(max-min+1)+min;
```

ne pozabi: x je v tem primeru že definirana celoštevilska spremenljivka

PRIMER: metanje kocke

6

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;

int main() {
    srand(time(0)); // določimo seme glede na uro

    int stPik = rand()%6 + 1;
    cout << "Na kocki je padlo stevilo ";
    switch (stPik) {
        case 1: cout << "ena."; break;
        case 2: cout << "dve."; break;
        case 3: cout << "tri."; break;
        case 4: cout << "stiri."; break;
        case 5: cout << "pet."; break;
        case 6: cout << "sest."; break;
    }
    return 0;
}
```