Lezione 06: String e random

La libreria standard `<string>`

La libreria `<string>` fornisce un tipo `string` ad **alta astrazione**, rispetto alle convenzioni del linguaggio C, che si basano su array di caratteri e puntatori.

Caratteristiche principali:

- **Mutable**: le stringhe possono essere modificate dopo la creazione.
- **Indicizzazione**: accesso diretto ai caratteri tramite `[]`.
- **Sottostringhe**: metodi come `substr()` e `replace()` semplificano la manipolazione.

Costruttori

```
string s0;                  // stringa vuota
string s1("Testo di esempio");   // inizializzazione da letterale
string s2(s1);               // copia
string s3(7, 'a');           // "aaaaaaa"
string s4(0);                // pericoloso!
string s5(nullptr);          // errore a runtime
string s6(p);                // dipende da p (puntatore)
string s7{"OK"};             // sicuro con letterale C-style
```

Operazioni comuni

```
s.size();      // numero di caratteri
s.length();    // identico a size()
s.empty();     // true se vuota
s.clear();     // svuota la stringa
s.front();     // primo carattere (s[0])
s.back();      // ultimo carattere
```

Concatenazione e modifiche

```
string compose(const string& name, const string& domain) {
   return name + '@' + domain;
}
auto addr = compose("anna.corazza", "unina.it");

string addNewline(string& s1, string& s2) {
   s1 = s1 + '\n';   // concatenazione
   s1 += '\n';       // operatore +=
}
```

Sottostringhe e sostituzioni

```
string name = "Niels Stroustrup";
string s = name.substr(6, 10);         // "Stroustrup"
name.replace(0, 5, "nicholas");        // "nicholas Stroustrup"
name[0] = toupper(name[0]);            // "Nicholas Stroustrup"
```

Numeri pseudo-casuali – Libreria `<random>`

La libreria `<random>` permette la generazione di numeri pseudo-casuali con un elevato grado di controllo.

Struttura:

1. **Motore di generazione** – genera numeri casuali grezzi.
2. **Distribuzione** – trasforma i numeri in base a una legge (uniforme, normale, esponenziale, ...).

```
#include <random>
#include <functional>
using namespace std;

using my_engine = default_random_engine;
```

```
using my_distribution = uniform_int_distribution<int>;

my_engine re {};                    // generatore (seed automatico)
my_distribution one_to_six {1, 6};  // distribuzione uniforme [1,6]
auto dado = bind(one_to_six, re);   // generatore personalizzato

int x = dado();                     // ottieni un numero casuale
```

seed()

- È possibile **inizializzare** il generatore con un *seed* personalizzato per ripetere gli stessi risultati (utile nei test).