JP, Wtorek, 20.11.2012, 16:15-17:45

Tym razem napiszemy coś bardziej praktycznego – klasę do generowania liczb pseudolosowych z trzech różnych rozkładów (jednorodnego, Gaussa oraz Poissona). Napiszemy również klasę do tworzenia histogramów z ciągów liczb (żeby sobie narysować wygenerowany rozkład) i poćwiczymy wczytywanie i zapisywanie do plików.

Klasa do generowania liczb losowych

Klasa będzie się nazywać `Random`. Jej plik nagłówkowy powinien zawierać:

- konstruktor domyślny – ustawia ziarno z zegara systemowego: `srand(time(NULL))`
- metoda `void SetSeed(int seed)` – ustawia ziarno
- metoda `double Uniform(double xmax)` - zwracająca liczbę pseudolosową z rozkładu jednorodnego z przedziału <0;xmax) – z domyślnie ustawionym przedziałem na <0;1)
- metoda `double Uniform(double xmin, double xmax)` – zwracająca liczbę pseudolosową z rozkładu jednorodnego z przedziału <xmin;xmax)
- metoda `double Gaus(double mean, double sigma)` – zwracająca liczbę pseudolosową z rozkładu Gaussa o zadanej średniej oraz odchyleniu standardowym (szczegóły implementacji **Uwaga 1!**)

Zestaw bibliotek niezbędnych do stworzenia generatora:
`cstdlib, ctime, cmath`
Należy użyć funkcji `rand()` - zwracającej liczbę całkowitą pseudolosową z rozkładu <0;RAND_MAX), gdzie `RAND_MAX` jest wielką liczbą - stałą zdefiniowaną w bibliotece standardowej.

Klasa do histogramowania

Klasa będzie się nazywać `Histogram`. Jej deklaracja powinna być następująca:

```
class Histogram
{
  std::string fname;
  double fxmin;
  double fxmax;
  unsigned int fnbins;
  unsigned int overflow; //zlicza wejscia powyzej fxmax
  unsigned int underflow; //zlicza wejscia ponizej fxmin
  unsigned int *frequency; //tablica histogramu

 public:
   Histogram(std::string name, double xmin, double xmax, unsigned
int nbins);
  Histogram(const Histogram& hist);
  ~Histogram();

  void Fill(double val);
  double GetBinWidth();
  void PrintHistogram();
};
```

- Destruktor usuwa tablicę `frequency`
- Metoda `void Fill(double val)` zwiększa wartość danego binu odpowiadającego wartości `val` histogramu o 1
- Metoda `double GetBinWidth()` zwraca szerokość binu w histogramie

- Metoda `void PrintHistogram()` wypisuje histogram na ekran, w formacie jak dla histogramu o nazwie `HistTest`:

```
Printing histogram HistTest
underflow: 2
overflow: 5
1: #
2: #####
3: #####
4: #######
5: ########
6: ########
7: #########
```

gdzie liczba `#` oznacza liczbę wejść w binie o zadanym numerze (biny numerujemy od 1!).

Do wykonania (robimy po kolei):

1. Tworzymy klasę `Random` z jej konstruktorem, metodą `void SetSeed(int seed)` oraz najprostszą metodą do generowania liczb pseudolosowych `double Uniform(double xmax)`
2. Tworzymy plik **mainRandom.cpp**, w którym sprawdzamy, czy nasza klasa `Random` działa (losujemy w pętli powiedzmy 30-40 liczb i patrzymy jakie mają wartości) **1.5 p.**
3. Jeśli działa, to zapisujemy nasz wynik do pliku **numbers.txt** używając strumieni wyjścia i ładując w programie bibliotekę `fstream`. Uwaga! Musimy w programie stworzyć obiekt typu `ofstream`. Plik zapisujemy w takiej postaci, by w każdej kolejnej linijce znajdowała się wylosowana liczba. **0.5 p.**
4. Tworzymy całą klasę `Histogram`. Następnie tworzymy plik **mainHistogram.cpp**, w którym będziemy otwierać zapisany wcześniej plik i wczytywać w pętli liczby i metodą `Fill` wpisywać do histogramu. Na końcu wypiszemy histogram na ekran metodą `PrintHistogram`. **2 p**.
5. Tworzymy pozostałe metody do generowania liczb losowych, zapisujemy dane za pomocą **mainRandom** do 3 różnych plików, następnie w jednym programie **mainHistogram** wczytujemy wszystkie 3 (obiekt typu `ifstream`), wypełniamy 3 histogramy i wypisujemy je na ekran. Na koniec przekierowujemy „output” uruchomionego programu **mainHistogram** do pliku **out.txt**. **1 p**.

Uwaga 1!
Algorytm do generowania liczb pseudolosowych z rozkładu Gaussa (metoda Box-Muller'a):

- U, V – dane liczby pseudolosowe z rozkładu jednorodnego <0;1)
- Liczba losowa z rozkładu Gausa o średniej μ i odchyleniu standardowym σ dana jest wtedy wzorem:
  $x = \mu + \sigma\sqrt{-2\ln(U)}\cos(2\pi V)$

Uwaga 2!
Przykład zapisywania do pliku:

```
#include <iostream>
#include <fstream>
using namespace std;

int main () {
  ofstream myfile;
  myfile.open ("example.txt");
```

```
  myfile << "Writing this to a file.\n";
  myfile.close();
  return 0;
}
```

Przykład wczytywania pliku:

```
#include <iostream>
#include <fstream>
using namespace std;

int main () {
  ifstream myfile;
  myfile.open ("example.txt");
  double val;
  while(myfile>>val)
  {
    //tu można cos zrobic z wartoscia val
  }
  myfile.close();
  return 0;
}
```