

Języki programowania, Czwartek 8.01.2015, 16:15-17:45

Tym razem napiszemy coś bardziej praktycznego – klasę do generowania liczb pseudolosowych z trzech różnych rozkładów (jednorodnego, Gaussa oraz Poissona).

Klasa do generowania liczb losowych – 1 p.

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

- pola `double fMin` i `double fMax` – przedział, z którego będą losowane liczby
- konstruktor domyślny – ustawia ziarno z zegara systemowego: `srand(time(NULL))` oraz `fMin` na 0 i `fMax` na 1
- Konstruktor z parametrami ustawiający pola `fMin` i `fMax`; `srand` ustawiamy analogicznie jak w konstruktorze bez parametrów.
- metoda `void SetSeed(int seed)` – ustawia ziarno `srand(seed)`
- metody „set” i „get” ustawiające pola `fMin` i `fMax`
- metoda `double Exec()` - metoda wirtualna, zwraca liczbę pseudolosową z rozkładu jednorodnego z przedziału `<fMin;fMax)`

Następnie tworzymy klasę `Gauss`, która dziedziczy z klasy `Random`. Powinna ona zawierać:

- pola typu `double fMean`, `double fSigma`, które przechowują parametry rozkładu Gaussa: średnią i odchylenie standardowe.
- konstruktor z parametrami ustawiający wszystkie pola jak
- metody typu „set” ustawiające oba pola i metody „get” zwracające oba pola
- metodę `double Exec()` - zwracającą liczbę pseudolosową z rozkładu Gaussa o zadanej średniej i odchyleniu standardowym (szczegóły implementacji patrz **Uwaga1!**)

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.

Piszemy program testujący działanie dwóch wyżej wymienionych klas – 2 p.

Tworzymy w funkcji `main` obiekt klasy `Random` (np. `Random rand1(-2, 3);`) z ustawionymi wartościami od -2 do 3. Losujemy kilka liczb pseudolosowych i wypisujemy na ekran.

Tworzymy w funkcji `main` obiekt klasy `Gauss` (np. `Gauss gaus1(100, 5);`) z ustawioną średnią na 100 i odchyleniem 5. Losujemy kilka liczb pseudolosowych.

Tworzymy wskaźnik na obiekt klasy `Random` i ustawiamy go na wcześniej stworzony obiekt tej klasy:

```
Random *rand2 = &rand1;
```

i wypisujemy kilka losowych wartości metodą `Exec()`. Następnie ten sam wskaźnik `rand2` ustawiamy na obiekt klasy `Gauss`:

```
rand2 = &gaus1;
```

i analogicznie wypisujemy kilka losowych wartości metodą `Exec()`.

W domu lub dodatkowo: można dodać klasę `Poisson` (czytaj **Uwaga2!**), zachowującą się podobnie do poprzedniej, wywołać na niej metodę `Exec()`.

Tworzymy klasę abstrakcyjną – 1 p.

Tworzymy klasę `Uniform`, do której „przerzucamy” z klasy `Random` pola `fMin`, `fMax` i algorytm metody `Exec()` losujący liczbę pseudolosową z rozkładu jednorodnego (powinna posiadać konstruktor z parametrami ustawiającymi `fMin` i `fMax`). Klasa dziedziczy z klasy `Random`, w której pozostało tylko pole `fSeed`.

W klasie `Random` tworzymy teraz metodę `Exec()` jako czysto wirtualną, tzn w pliku `Random.h`:

```
virtual double Exec() = 0;
```

Poprawiamy następnie program główny w taki sposób, by się kompilował.

Jeśli klasa deklaruje jedną ze swoich funkcji jako `virtual`, wówczas jej destruktory deklarujemy także jako `virtual`. Skoro w klasie deklarujemy jakąś funkcję wirtualną, to znaczy, że na obiekty klas pochodnych zamierzamy czasem mówić jak na obiekty klasy podstawowej, co przy późniejszym niszczeniu obiektów mogłoby być problemem – nie zwalnialiśmyby pamięci dla niektórych składników klas pochodnych.

Obsługa plików tekstowych – 1 p.

W programie zapisujemy kilka wylosowanych liczb z rozkładu Gaussa do pliku tekstowego.

Przykład zapisywania do pliku:

```
#include <fstream>
```

```
...
```

```
ofstream ofile;  
ofile.open("file.txt");  
ofile<<"aaa"<<123<<endl;  
ofile.close();
```

Następnie wczytujemy zapisany plik i wypisujemy wszystkie liczby na ekran. Przykład wczytywania liczb z pliku:

```
#include <fstream>
```

```
...
```

```
ifstream ifile;  
int val;  
ifile.open("file.txt");  
while(ifile>>val)  
{  
    cout<<"val: "<<val<<endl;  
}  
ifile.close();
```

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 Gaussa o średniej μ i odchyleniu standardowym σ dana jest wtedy wzorem:

$$x = \mu + \sigma \sqrt{-2 \ln(U)} \cos(2\pi V)$$

Uwaga 2!

Algorytm do generowania liczb pseudolosowych z rozkładu Poissona o średniej λ :

- $x_0, x_1, \dots$ - ciąg liczb pseudolosowych z rozkładu jednorodnego
- Jeżeli $x_0 x_1 \dots x_k < e^{-\lambda}$, to k jest liczbą z rozkładu Poissona.