

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

Tworzymy klasę abstrakcyjną (ATD – abstrakcyjny typ danych) – 1 p.

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

- pole `int fSeed` do przechowywania tzw. ziarna (ang. seed) – jeśli inne niż `time (NULL)`
- konstruktor domyślny – ustawia ziarno z zegara systemowego: `srand (time (NULL))`
- konstruktor z 1 parametrem typu `int seed` – ustawia ziarno zegara systemowego `srand (seed)` i przypisuje tą wartość również do pola `fSeed`
- metoda czysto wirtualna `double Exec ()`

Klasa do generowania liczb losowych z rozkładu jednorodnego – 1 p.

Klasa będzie się nazywać `Uniform` i dziedziczyć z klasy abstrakcyjnej `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))` (używając konstruktora klasy nadrzędnej w liście inicjalizacyjnej) oraz `fMin` na 0 i `fMax` na 1
- Konstruktor z parametrami ustawiający pola `fMin`, `fMax` i `fSeed` (do ustawienia ziarna używamy konstruktora z parametrami klasy nadrzędnej w liście inicjalizacyjnej)
- metody „set” i „get” ustawiające pola `fMin` i `fMax`
- metoda `double Exec ()` - metoda wirtualna, zwraca liczbę pseudolosową z rozkładu jednorodnego z przedziału `<fMin;fMax)`

Jeśli starczy czasu, 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 ()` - metoda wirtualna 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 (po załączeniu wszystkich bibliotek możemy jej używać w programie).

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

Tworzymy w funkcji `main` obiekt klasy `Uniform` (np. `Uniform uni1 (-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 *rand = &uni1;
```

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

```
rand = &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 ()`.

Wskazówka: wszystkie metody czysto wirtualne muszą istnieć w klasach pochodnych.

Klasa, która ma co najmniej jedną funkcję czysto wirtualną nazywamy **klasą abstrakcyjną**.

Obsługa plików tekstowych – 1 p.

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

```
#include <fstream> //naglowek zawierajacy funkcje C++ do operacji na plikach
```

```
...
```

```
ofstream ofile;  
ofile.open("file.txt"); //"file.txt" jest tutaj typu char*  
ofile<<"aaa"<<123<<endl; //zapisze w pliku ciag znakow aaa123  
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) //petla od początku do konca pliku  
{  
    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.