
JĘZYK PROGRAMOWANIA C++

LICZBY LOSOWE

Instytut Informatyki Uniwersytetu Wrocławskiego

Paweł Rzechonek

Liczba losowa to liczba otrzymana jako rezultat działania określonego mechanizmu losującego (na przykład przy rzucaniu kostką do gry, tasowaniu kart, ciągnięciu losów z urny itp.). Można je także uzyskiwać za pomocą specjalnie skonstruowanych urządzeń albo odpowiednich programów komputerowych zwanych generatorami liczb losowych.

Otrzymywanie liczb prawdziwie losowych, a więc takich, które faktycznie nie ukrywają w sobie żadnej prawidłowości, jest bardzo trudne. Dlatego często mówi się o liczbach pseudolosowych, czyli takich, których rozkład ma pewne ukryte regularności, ale nieistotne z punktu widzenia technicznego.

Generatory liczb losowych są ważnymi współczesnymi narzędziami badawczymi, używanymi w wielu dziedzinach, między innymi w kryptografii czy w statystyce. Wynikiem działania generatora liczb losowych zwykle są liczby z przedziału $[0, 1)$ o rozkładzie jednostajnym. Z generatora o takim rozkładzie można uzyskać generator o dowolnym innym rozkładzie obliczając odwrotną dystrybuantę pożądanego rozkładu z wyników pierwszego generatora.

Zadanie 1.

Zdefiniuj klasę `osoba` z polami typu `string` do zapamiętania imienia i nazwiska (pola `imie` i `nazwisko`) oraz z polem typu `time_point` do zapamiętania daty urodzenia (pole `ur`). W klasie tej powinien być zdefiniowany konstruktor do zainicjalizowania wszystkich wymienionych pól obiektu wartościami przekazanymi przez argumenty. Zaimplementuj też pełną semantykę kopiowania w klasie `osoba`.

Następnie stwórz plik tekstowy z danymi osobowymi, którymi będzie można zainicjalizować kolekcję osób: w pierwszej linii niech będzie zapisana liczba osób a w kolejnych liniach informacje personalne (imię i nazwisko). Datę urodzenia dla każdej osoby wygeneruj losowo z rozkładem jednostajnym z zakresu od 1 września 1939 do 31 maja 2017 roku. Na przykład:

```
9
Adam Kowal
Iwona Rybak
Bartosz Nowak
Helena Puc
Celina Wolska
Grzegorz Puc
Daniel Rybak
Franciszk Nowak
Ewa Nowak
```

Na koniec napisz program, który wczyta dane z pliku z danymi osobowymi do pamięci, umieszczając wszystkie informacje w kolekcji `list<osoba>` a potem posortuje je najpierw względem nazwiska i imienia a potem względem daty urodzenia (po każdym sortowaniu wypisz wszystkie osoby w kolekcji). Jako parametru porównywania w sortowaniu użyj `lambda`.

Zadanie 2.

Napisz program, który zapisze do pliku ciąg n liczb całkowitych z zakresu od 0 do k z rozkładem dwumianowym. W pierwszej linijce pliku zapisz wartość n a w następnych liniach kolejne wartości ciągu. Zarówno wartość n , k jak i elementy ciągu wylosuj używając silnika `mt19937`.

Następnie napisz kolejny program, który wczyta ciąg z pliku z danymi do pamięci, umieszczając wszystkie informacje w kolekcji `list<int>`. Zastosuj do tej kolekcji algorytm `for_each` z takim obiektem funkcyjnym (z takim stanem wewnętrznym), który pozwoli potem na wskazanie wartości minimalnej, maksymalnej i średniej w tej kolekcji.

Zadanie 3.

Do kolekcji `deque<double>` wpisz kilkadziesiąt losowo wygenerowanych liczb — liczby mają być wygenerowane z rozkładem normalnym (rozsądnie dobierz parametry tego rozkładu).

Następnie zdefiniuj predykat, który sprawdzi, czy podana wartość jest dodatnia, i mniejsza od 3. Predykat ten skonstruuj z predefiniowanych obiektów funkcyjnych. Predykatu tego użyj do zliczenia wszystkich elementów kolekcji spełniających podany warunek (algorytm `count_if`) oraz do wyznaczenia pierwszej wartości spełniającej ten warunek (algorytm `find_if`).

Zadanie 4.

Do kolekcji `deque<int>` wpisz kilkadziesiąt losowo wygenerowanych liczb — liczby mają należeć do zbioru $\{0 \dots 10\}$ i mają być wygenerowane z rozkładem dwumianowym.

Następnie usuń z tej kolekcji algorytmem `remove_if` wszystkie wartości mniejsze od 2 i większe od 8. W algorytmie tym wykorzystaj odpowiednio zdefiniowaną predykatową funkcję `lambda`.

Na koniec za pomocą algorytmu `count_if` zlicz wszystkie elementy w kolekcji, które są większe od 4 i mniejsze od 6. W algorytmie tym wykorzystaj odpowiednio zdefiniowaną predykatową funkcję `lambda`.

Uwaga.

Każdy program zapisz w oddzielnym pliku. Nie wyodrębniaj w tych krótkich programach plików nagłówkowych.

Elementy w programie, na które należy zwrócić szczególną uwagę.

- Wykorzystanie różnych generatorów losowych i różnych rozkładów prawdopodobieństwa.
- Wykorzystanie algorytmów i kolekcji standardowych.
- Posługiwanie się `lambdami`.