

kurs języka C++

kolejka punktów

Instytut Informatyki
Uniwersytetu Wrocławskiego

Paweł Rzechonek

Prolog

Kolejka to bufor na dane, do którego można wkładać nowe elementy oraz wyciągać elementy, ale tylko w kolejności zgodnej z czasem ich włożenia – z kolejki można usunąć tylko ten element, który został do niej włożony najwcześniej. Odpowiednikiem tej struktury w życiu codziennym może być kolejka klientów do kasy w sklepie – klient, który ustawił się w kolejce jako pierwszy zostanie jako pierwszy obsłużony. Stąd się wzięła nazwa takiego bufora: FIFO (ang. *First In, First Out*), czyli „kto pierwszy przyszedł ten pierwszy wyjdzie”.

Zadanie

Zdefiniuj klasę punkt, która będzie reprezentowała punkt na płaszczyźnie euklidesowej. Klasa punkt powinna zawierać dwa pola x i y typu double do pamiętania współrzędnych punktu. Zdefiniuj konstruktor nadający początkowe współrzędne punktowi oraz konstruktor bezargumentowy umieszczający punkt w początku układu współrzędnych. Pamiętaj o hermetyzacji, aby ukryć stan każdego obiektu. W klasie tej umieść także gettery do odczytu współrzędnych punktu i settery do ustanowienia nowych współrzędnych.

Zdefiniuj klasę kolejka, która będzie strukturą typu FIFO – element, który został do tej struktury dodany najwcześniej, będzie z niej wyciągnięty najszybciej. Struktura ta ma służyć do przechowywania punktów na płaszczyźnie typu punkt.

Sama funkcjonalność kolejki ma być bardzo prosta:

1. wkładamy punkt do kolejki – funkcja składowa `void wstaw(punkt);`
2. wyciągamy punkt z kolejki – funkcja składowa `punkt usun();`
3. podglądamy punkt z początku kolejki – funkcja składowa `punkt zprzodu();`
4. pytamy o liczbę wszystkich elementów w kolejce – funkcja składowa `int dlugosc();`

Kolejkę zaimplementuj na tablicy utworzonej dynamicznie na stercie w trakcie działania konstruktora (za pomocą operatora `new[]`, na przykład `kol = new punkt[pojemnosc]`). W destruktorze należy zwolnić pamięć przydzieloną dla tej tablicy (za pomocą operatora `delete[]`, na przykład `delete[] kol`). Implementując kolejkę na tablicy najlepiej jest wykorzystać zawinięcie tablicy (tablica tworzy coś w rodzaju pętli – następnikiem ostatniego elementu jest pierwszy element), co istotnie ułatwi obsługę tej struktury.

Pojemność kolejki ma zostać określona w konstruktorze – zdefiniuj więc prywatne pole `pojemnosc` typu `int`, w którym będzie pamiętany maksymalny rozmiar kolejki. Będziesz też potrzebować informacji o miejscu gdzie kolejka się zaczyna oraz o liczbie elementów aktualnie przechowywanych w kolejce – zdefiniuj zatem prywatne pola `pocz` oraz `ile` typu `int`, w których będziesz pamiętać odpowiednio miejsce pierwszego elementu oraz liczbę wszystkich elementów przechowywanych w kolejce.

```
class kolejka {
    int pojemnosc, pocz = 0, ile = 0;
    punkt *tab;
    // ...
};
```

Kolejka ma posiadać pięć konstruktorów: konstruktor zadaną pojemnością, konstruktor bezparametrowy i jednocześnie delegatowy (domyślna pojemność kolejki to 1), konstruktor inicjalizujący kolejkę za pomocą listy punktów typu `initializer_list<punkt>`, konstruktor kopiujący i przenoszący. Aby uzupełnić semantykę kopiowania i przenoszenia zdefiniuj odpowiednie operatory przypisania (przypisanie kopiujące i przenoszące).

Na koniec napisz program rzetelnie testujący działanie kolejki. W programie testującym uruchom wszystkie funkcje składowe kolejki w różnych sytuacjach (również w tych wyjątkowych). Dokładnie przetestuj kopiowanie i przenoszenie kolejek. Rezultaty wywołanych funkcji w tym konstruktorów i destruktora zaprezentuj na standardowym wyjściu i na standardowym wyjściu dla błędów (komunikaty z konstruktorów i destruktora kolejki oraz z operatorów przypisań).

Uwaga

Podziel program na pliki nagłówkowe z definicjami klas i źródłowe z definicjami funkcji składowych zadeklarowanych w klasach. Funkcję `main()` z testami umieść w osobnym pliku źródłowym.

Ważne elementy programu

- Obiekt kolejki ma być inicjalizowany na kilka różnych sposobów: konkretną pojemnością, domyślnie (konstruktor delegatowy), przez skopiowanie z innej kolejki (konstruktor kopiujący), za pomocą listy wartości początkowych, za pomocą przeniesienia zawartości z kolejki tymczasowej (konstruktor przenoszący).
- Obiekt kolejki ma być kopiowalny (przypisanie kopiujące i przenoszące).
- Zgłaszanie wyjątków w konstruktorach i funkcjach składowych.
- Program testujący w funkcji `main()`.