

①. Доступ до членів класу по посиланню.

②. Прості черги.

③. Метод обміну.

①

Посилання (змінна-посилання) на об'єкт класу може бути членом даних іншого класу. При оголошенні змінної-посилання в класі, ця змінна має бути ініціалізована одразу в спеціально розробленому конструкторі. У цьому випадку для змінної-посилання класу пам'ять виділяється динамічно. Також важливо пам'ятати, що при оголошенні посилання на об'єкт деякого класу, обов'язково потрібно ініціалізувати це посилання деяким значенням, наприклад, у спеціально розробленому конструкторі за замовчуванням. Якщо не вказати код ініціалізації посилання в конструкторі, то компілятор видасть помилку.

②

Взагалі черга являє собою лінійний список, доступ до елементів якої здійснюється за принципом FIFO (first in, first out – першим зайшов, першим вийшов). Тому, першим із черги видаляється елемент, який був записаний до черги першим, потім – елемент, що був записаний до черги другим, і т. д. Для черги такий метод доступу і збереження даних являється єдиним.

Прості черги мають досить широке використання на практиці. Наприклад при моделюванні процесів реального часу, для диспетчеризації завдань операційної системи або для буферизації операцій вводу-виводу.

Така черга повинна реалізовувати наступні операції:

1. додавання елемента даних у кінець черги;
2. зчитування та вилучення елемента даних з початку черги.

Проста черга має деякі недоліки, а саме: фіксований розмір, який не можна змінити в процесі роботи з чергою, можливість втрати даних при переповненні черги, оскільки не можна продовжувати запис даних до заповненої черги, поки не будуть прочитані всі елементи. Лише після того, як буде прочитаний останній елемент черги, до неї можна знов записувати дані, починаючи з першого елемента. Цього можна уникнути, створюючи, в разі необхідності, нові черги в динамічному режимі. Але при такому підході не досить ефективно використовується пам'ять. Можна розробити безрозмірну чергу на основі списку, але це досить складно.

Сортування за допомогою обміну базується на процесі порівняння і при необхідності обміну місцями двох сусідніх елементів масиву. Ці операції повторюються доти, доки не буде упорядковано весь масив. Треба зазначити, що після першого проходу по всьому масиву максимальний елемент переміщається в крайнє праве положення, і на наступному етапі немає сенсу перевіряти весь масив. Тому на практиці при першому проході перевіряють елементи з номерами від 1 до n (останнього), на другому від 1 до $(n-1)$ і т.д.