

## Домашнє завдання № 8

**Динамічні структури даних : стек, черга, черга з пріоритетами.**

### ***Завдання 1.***

Створити шаблон класу PriorityQueue(Черга з пріоритетами, динамічно розширювана).

Вважати, що більший елемент має більший пріоритет. Протестувати на типах int, string, Drib.

### ***Завдання 2.***

Виконати на вибір одне із завдань.

Можна використати у завданні 2.1 шаблон класу Стек. Можна використати у завданні 2.2 раніше побудований шаблон класу Черга з пріоритетами(із завдання 1).

#### ***Завдання 2.1.***

Напишіть програму з трьома класами - **автомат, магазин, набій**.

Кожен **набій** пронумерований (нумерація послідовна і при запуску програми починається з випадкового числа) і вміє вистрілювати і влучати в мішень з імовірністю 70%. Кожен набій у програмі має унікальний номер.

**Магазин** реалізований у вигляді стеку, містить 30 набоїв і вміє:

- о заряджати набій (отримує його «готовим»)

- о віддавати набій (верхній)

**Автомат** вміє:

- о прийняти магазин(тобто можна змінювати магазини у автоматі)

- о стріляти - попросити магазин дати 1 набій, і, якщо дав, попросити набій вистрелити (набій при цьому скаже, чи влучив, а автомат передасть результат «нагору» )

При пострілі в консоль має виводитись повідомлення такого типу:

*Автомат вистрелив кулю 5, і промахнувся.*

*Автомат вистрелив кулю 6, і влучив.*

#### ***Завдання 2. 2.***

Декілька користувачів мережі друкують свої документи на одному мережевому принтері. В один момент часу принтер може друкувати лише один документ. На друк документа іде певний час, для кожного документа свій ( rand() ). Протягом усього цього часу принтер «зайнятий» і не може приймати нового завдання.

Якщо, поки принтер «зайнятий», якийсь користувач відправляє документ на друк, то цей, новий, документ ставиться у чергу друку.

Як тільки принтер завершить друк документа, він стає «вільний» і може прийняти інший документ і почати його друкувати.

Документи різних користувачів мають різний пріоритет. Пріоритет документів «Директора» вищий від пріоритету документів «Бухгалтера», а пріоритет документів «Бухгалтера» вищий від пріоритету документів «Клерка» і т. д.

Отже, коли у черзі є документи з різними пріоритетами, то на принтер з черги спершу виходять документи з вищим пріоритетом.

Треба реалізувати клас **Spooler**, котрий є пріоритетною чергою і може приймати документи з пріоритетом, та віддавати їх (пріоритетні в першу чергу)

Черга впорядковується при постановці -- новий документ вставляється перед першим документом з нижчим пріоритетом, або в хвіст, якщо такого нема.

**Документ** -- окремий клас (Document), котрий «знає»:

- свою назву (Doc1.doc -- можна використати String )
- свій розмір ( 23456 KB -- int ) -- від розміру залежатиме час його друку принтером
- вміє : повідомити це все.

**Принтер** -- клас Printer, котрий знає:

- екземпляр Spooler-а, у якого запитуватиме документи
- документ, котрий друкує в даний момент
- скільки часу залишилося до завершення, або момент завершення

Треба зробити меню, щоб ставити документ з різним пріоритетом у чергу, а потім запустити їх на друк