

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп'ютерні інформаційні технології»

**Лабораторна робота №4**

**з дисципліни «Алгоритми та структури даних»**

**на тему: «Робота з шаблонами(generic java)»**

Виконав:  
студент гр.ПЗ1911  
Сафонов Д. Є.  
Прийняла:  
Куруп'ятник О. С.

Дніпро, 2021

**Тема.** Робота з шаблонами(generic java).

**Мета.** Ознайомитися з поняттям узагальнення, варіантами його застосування. Отримати практичні навички роботи з колекціями як різновидом узагальнень.

**Завдання.** Виконайте моделювання ситуації,описаної в індивідуальному завданні,з використанням структур JavaCollectionFramework. Обґрунтуйте вибір структури даних, виконайте аналіз можливих альтернатив.

**Індивідуальне завдання.**

Варіант	Ситуація	Значення параметризованого типу
6	Снігуронька записує бажання дітей, які шикуються у чергу. Бажання можуть бути таких видів: іграшки, гаджети, одяг, смаколики. Дід Мороз виконує бажання по одному у порядку надходження. Порахувати скільки бажань кожного виду вже виконав Дід.	Бажання

**Текст програми.** [github](#)

### **Вибір та обґрунтування колекції.**

java.util.LinkedList<E> - реалізує інтерфейс java.util.Queue<E>, є двозв'язним списком, тож складність додавання та видалення елементів у обидва кінці є сталою, інші операції не потрібні, тож цей клас є найкращим вибором.

Інші класи, які були розглянуті(які реалізують інтерфейс черги), але не підійшли:

- класи із Array у назві - через те, що вони засновані на масиві, вони або мають обмежений розмір, або виділення пам'яті під нові елементи дорожче ніж у обраному класі.
- класи із Concurrent у назві та LinkedTransferQueue - нічим не краще ніж обраний клас у даній задачі, мають додатковий кошттовний функціонал для взаємодії з багатьма потоками.
- класи із Blocking у назві та SynchronousQueue, DelayQueue - класи із специфічним функціоналом для асинхронних програм.
- PriorityQueue - черга із пріоритетом, додатковий непотрібний функціонал.

## Висновки.

Була розроблена програма, яка спочатку у циклі генерує різні бажання, а потім у іншому циклі рахує їх за типами. Ця програма має лише один потік, авжеж можна було розробити програму із двома потоками, де один потік генерує значення(можна обмежити кількість бажань у коді, або спитати у користувача на початку програми, або додати третій потік, який буде очікувати поки користувач не зупинить потік бажань), а другий потік буде їх рахувати. Подібна модель буде більш реалістичною, і для неї знадобиться синхронна черга(SynchronousQueue). Але програму можна ускладнити до нескінченності, і на якійсь ітерації вже не вистачить стандартного функціоналу, тож розроблена програма моделює задану ситуацію з найбільшою ефективністю.

```
see 'help', to check out available commands
###CHRISTMAS###: help
Commands:
    quit
    receive
    ls - print queue
    sz - print amount of elems in queue
    stats
    wish "<type>" "<receiver>", type = TOY | GADGET | CLOTH | FOOD
###CHRISTMAS###: receive
no wishes available, try to 'wish' first!(See 'help')
###CHRISTMAS###: ls

###CHRISTMAS###: sz
0 wishes available
###CHRISTMAS###: stats
Toys: 0
Gadgets: 0
Clothes: 0
Food: 0

###CHRISTMAS###: help
Commands:
    quit
    receive
    ls - print queue
    sz - print amount of elems in queue
    stats
    wish "<type>" "<receiver>", type = TOY | GADGET | CLOTH | FOOD
###CHRISTMAS###: wish TOY me
<wish TOY me> is incorrect command, try 'help'
###CHRISTMAS###: wish "TOY" "me"

###CHRISTMAS###: receive
received Wish: { WishType: TOY, receiver: me, received: true }
###CHRISTMAS###: stats
Toys: 1
Gadgets: 0
Clothes: 0
Food: 0

###CHRISTMAS###: []
```

Рисунок 1(Демонстрація роботи)