# Лабораторна робота №4.

**Мета:**

● Вивчення принципів параметризації в ​ Java ​ .

● Розробка параметризованих класів та методів.

● Розширення функціональності параметризованих класів.

# Вимоги

# 1. Створити власний клас-контейнер, що параметризується (​ Generic Type​ ), на основі ​ зв’язних списків​ для реалізації колекції domain-об’єктів з лабораторної роботи №10 (Прикладні задачі. Список №2. 20 варіантів)

# 2. Для розроблених класів-контейнерів забезпечити можливість використання їх об’єктів у циклі ​ foreach​ в якості джерела даних.

# 3. Забезпечити можливість збереження та відновлення колекції об’єктів: 1) за допомогою стандартної серіалізації; 2) не використовуючи протокол серіалізації.

# 4. Продемонструвати розроблену функціональність: створення контейнера, додавання елементів, видалення елементів, очищення контейнера, перетворення у масив, перетворення у рядок, перевірку на наявність елементів.

# 5. Забороняється використання контейнерів (колекцій) з ​ Java Collections Framework​ .

# 6. Розробити параметризовані методи (​ Generic Methods​ ) для обробки колекцій об’єктів згідно (Прикладні задачі. Список №2. 20 варіантів).

# 7. Продемонструвати розроблену функціональність (створення, управління та обробку власних контейнерів) в діалоговому та автоматичному режимах. a. Автоматичний режим виконання програми задається параметром командного рядка ​ -auto​ . Наприклад, ​ java ClassName -auto​ . b. В автоматичному режимі діалог з користувачем відсутній, необхідні данні генеруються, або зчитуються з файлу.

# Розробник

Горностай Богдан , КН-108

# Задача

1. 8. Автостанція. Запис в розкладі: номер рейсу; час відправлення; дні тижня; кількість вільних місць; маршрут - необмежений набір значень у вигляді “назва станції, час прибуття”.

# Опис програми

Програма створює вашу власну аудіотеку за допомогою класу, можна додавати, чистити, виписувати, серіалізувати та десеріалізувати ваші аудіотеки у файл вибраний користувачем

# Засоби ООП

Були використані різні класи та методи, структури даних та модифікатори доступу.

# Ієрархія та структура класів

# [Main](https://github.com/BohdanHornostai/OOP/blob/master/Lab3/Main.java) – реалізує меню програми

# [AutoStation](https://github.com/BohdanHornostai/OOP/blob/master/Lab3/AutoStation.java) – domain-об’єкт

[**PathGetter**](https://github.com/BohdanHornostai/OOP/blob/master/Lab3/PathGetter.java) - повертає шлях збереження файлу та демонструє вміст

[RoutePoint](https://github.com/BohdanHornostai/OOP/blob/master/Lab3/RoutePoint.java) та [SchedulePost](https://github.com/BohdanHornostai/OOP/blob/master/Lab3/SchedulePost.java) – підкласу класу [AutoStation](https://github.com/BohdanHornostai/OOP/blob/master/Lab3/AutoStation.java)

Клас LinkedList спеціально розроблений за для виконання умови «Лабораторна №4». Представляє собою типовий зв’язний список.

# ВИСНОВКИ

У ході роботи ми вивчили принципи параметризації в Java та розробили параметризований LinkedList, також розширили його функціональність