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: Гринишин Анастасія , КН-108, номер варіанту індивідуального завдання – 9.

Задача:

Параметризація в Java. Обробка параметризованих контейнерів

Ієрархія та структура класів:

1.Клас Main, який містить 1 функцію – main.

2.Клас Filego, який містить 2 функції – doFile, make_info.

3.Клас Demain,який містить 3 поля – numberFlight, date, numberFlight і їхні гетери та сетери.

4.Клас Station, який містить 4 поля – nameStation, dateArrival, dateDeparture, freeSeat.

5.Клас ConstructorsForXML , який містить 2 функції – WriteParamXML, read.

6.Клас SimpleArray, якиц є контейнером.

7.Інтерфейс Simple.

8.Клас ArrayIterator, який містить 2 функції – hasNext, next.

Фажливий фрагмент коду:

```
            }

        }
        if (s4.equals("2")) {
            System.out.println("Ви хочете вибрати файл, куди добавити інформації?(T/H):");
            s3 = in.nextLine();
            if (s3.equals("T")) {
                FILENAME = file.doFile();
            }
            train.clear();
            train = constructorsForXML.read(FILENAME);

            System.out.println("Введіть кількість маршрутів: ");
            int K = in.nextInt();


            for (int k = 0; k < K; k++) {
                Demain demain = new Demain();
                train.add(demain);
                M++;
            }
        }
        if (s4.equals("3")) {

            System.out.println("Ви хочете вибрати файл, звідки будете дивитися інформацію?(T/H):");
            s3 = in.nextLine();

            if (s3.equals("T")) {
                FILENAME = file.doFile();
            }
            System.out.print("Дивитися з файлу .xml чи .out(1/2): ");
            s3=in.nextLine();
            Simple<Demain> train2 = new SimpleArray<>();
            if(s3.equals("1")) {
                train2.clear();
                train2 = constructorsForXML.read(FILENAME);
            }
            else {
                FileInputStream fis = new FileInputStream( name: "temp.out");
                ObjectInputStream oin = new ObjectInputStream(fis);
                train2 = (Simple<Demain>) oin.readObject();
            }
           // System.out.println(train2.get(0).getNumberFlight());
            String[] strings = file.make_info(train2);
            for (String s : strings) {
                System.out.println(s);
            }
            //System.out.println(train2.size());

        }
        if (s4.equals("4")) {
                System.out.println("Ви хочете вибрати файл?(T/H):");
                s3 = in.nextLine();
                if (s3.equals("T")) {
                    FILENAME = file.doFile();
                }
            System.out.print("Зберегти в .xml чи .out файл?(1/2): ");
                s3=in.nextLine();
```

Висновок

У ході даної роботи я навчилася зберігати дані 2 способами (сереалізація і власна). Розробила власний контейнер який складався з головного класу , інтерфейсу та додаткового класу.