

# Paradygmaty programowania – ćwiczenia

## Lista 8

1. (OCaml) (Java) Dany jest następujący interfejs dla kolejek.

```
public interface MyQueue<E> {  
    public void enqueue(E x) throws FullException;  
    public void dequeue();  
    public E first() throws EmptyException;  
    public boolean isEmpty();  
    public boolean isFull();  
}
```

Napisz dwie klasy publiczne dla wyjątków `FullException` i `EmptyException`.

Napisz klasę generyczną, implementującą interfejs `MyQueue`, w której kolejka jest reprezentowana przez tablicę cykliczną (patrz zadanie 2 z listy 7). Użyj kolekcji `ArrayList`.

Przeprowadź test na małej kolejce (np. o rozmiarze 3), którą całkowicie zapełnisz.

2. Przeanalizuj następujący program w Javie. Czy ten program się skompiluje? Jeśli nie, to dlaczego i jak go poprawić (bez zmieniania argumentów metod)?

```
public class Test {  
    int zawartość = 0;  
    static void argNiemodyfikowalny(final Test zmienna) {  
        zmienna.zawartość = 1;  
        zmienna = null;  
    }  
    static void argModyfikowalny(Test zmienna) {  
        zmienna.zawartość = 1;  
        zmienna = null;  
    }  
    public static void main(String[] args) {  
        Test modyfikowalna = new Test();  
        final Test niemodyfikowalna = new Test();  
        // tutaj wstaw instrukcje  
    }  
}
```

Co i dlaczego zostanie wyświetlone, jeśli wiersz „// tutaj wstaw instrukcje” zastąpimy następującymi instrukcjami:

- a) `argNiemodyfikowalny(modyfikowalna);`  
`System.out.println(modyfikowalna.zawartość);`
- b) `argNiemodyfikowalny(niemodyfikowalna);`  
`System.out.println(niemodyfikowalna.zawartość);`

```
c) argModyfikowalny(modyfikowalna);  
   System.out.println(modyfikowalna.zawartość);  
d) argModyfikowalny(niemodyfikowalna);  
   System.out.println(niemodyfikowalna.zawartość);
```