

ANS w Nowym Sączu	
Niezwadność systemów informatycznych	
Sprawozdanie 1	
Grupa	P1
Imię i nazwisko	Paweł Kolawa

Zadanie 1

Zaimplementuj klasy: stos, kolejka na bazie następnego interfejsu polimorficznego.

```
abstract class Container
{
    protected int pointer = -1;
    protected int[] buffer = new int[10];

    public abstract int pop();

    public abstract int getCount();

    public abstract bool isEmpty();

    public abstract void push(int value);

    public abstract void clear();

    public abstract bool isFull();
}
```

```
public override int pop()
{
    if (this.pointer > -1)
    {
        this.pointer -= 1;
        return this.buffer[pointer + 1];
    }
    else
    {
        return 0;
    }
}

public override void push(int value)
{
    if (pointer < 10)
    {
        this.pointer += 1;
        this.buffer[pointer] = value;
    }
}
```

Metoda Pop – usuwa pierwszy element z kolejki

Metoda Push – wypycha elementy do kolejki

Zadanie 2

Do powyższej implementacji dołóż możliwości sprawdzania rozmiaru kolekcji, czyszczenia jej itp (metody Clear(), GetCount(), IsEmpty(), IsFull()). Staraj się to wykonać w optymalny sposób, korzystając z dziedziczenia i polimorfizmu.

```
public override void clear()
{
    this.pointer = -1;
}

public override int getCount()
{
    return this.pointer + 1;
}

public override bool isFull()
{
    if (pointer == 9)
        return true;
    else
        return false;
}

public override bool isEmpty()
{
    if (pointer > -1)
        return false;
    else
        return true;
}
```

Metoda Clear() – wyzerowanie, usunięcie wszystkich elementów

Metoda GetCount() – zlicza ilość elementów

Metoda IsFull() – sprawdza czy stos lub kolejka są pełne

Metoda IsEmpty() – sprawdza czy stos lub kolejka są puste