Algorytmy i złożoność obliczeniowa

Przygotowanie do laboratorium 3: Stosy. Proste algorytmy wykorzystujące stosy.

1. **Koncepcja stosu.**
   Stos jest jednym z rodzajów struktur danych. Kolejne elementy (dane) są układane na wierzchołku stosu. Usunięty może być tylko element z wierzchołka stosu. Dobrym przykładem obrazującym działanie stosu jest stos książek na biurku. Możemy dołożyć kolejną książkę kładąc ją na wierzchołku, po czym to właśnie nowo dołożona książka staje się wierzchołkiem stosu. Jeśli chcemy z takiego stosu wziąć książkę, to możemy wziąć tylko książkę, która leży na wierzchu. Jeśli chcemy sięgnąć po inną książkę, musimy kolejno zdejmować książki z wierzchołka.
2. **Operacje na stosie.**
   push – dołożenie elementu na stos
   pop – ściągnięcie elementu ze stosu (i zwrócenie jego wartości)
   isEmpty – sprawdzenie, czy stos jest pusty
   Często implementacje stosu dostarczają dodatkowych operacji, jak np.
   peek – sprawdzenie wartości elementu na wierzchołku stosu bez jego zdejmowania

3. **Interfejs IStos – stos przechowujący liczby całkowite.**

```
public interface IStos {
    public void push(int i);  // dodaje i na stos
    public int pop();          // zdejmuje element, zwraca wartość z wierzchołka
    public int peek();         // zwraca wartość z wierzchołka stosu
    public boolean isEmpty(); // sprawdza, czy stos jest pusty
    public void print();       // wypisuje na ekran zawartość stosu
    public void clear();       // usuwa wszystkie elementy ze stosu
}
```

Zadania.

1. Zaimplementować interfejs **IStos** w oparciu o tablicę liczb całkowitych o rozmiarze 10. W tym celu proszę utworzyć klasę **TStos** zawierającą tablicę do przechowywania elementów stosu oraz licznik elementów.
2. Dodać wyrzucanie wyjątków w sytuacjach, które prowadzą do niespójnego stanu stosu (np. próba dołożenia elementu na całkowicie zapełniony stos, albo próba zdjęcia elementu z pustego stosu). Wyjątek rzucamy za pomocą instrukcji throw, np. *throw New Exception(„Stos pełny!”);*
3. W klasie TStos zaimplementować dwa dodatkowe konstruktory: jeden umożliwiający podanie rozmiaru tablicy (przyjmuje int jako parametr), drugi konstruktor (tzw. Konstruktor kopiujący) ma przyjmować jako parametr inny obiekt typu TStos i ma skopiować jego zawartość do nowo tworzonego obiektu.
4. Utworzyć klasę TStosDyn, która implementuje interfejs IStos i również używa tablicy, ale w momencie dodania danych do pełnego stosu wewnętrzna tablica jest powiększana dwukrotnie, a elementy są przepisywane.