

BIL207 VERİ YAPILARI

4. Hafta

Yığın/Yığıt (Stacks) Örnek Programlar

Doç. Dr. Sercan YALÇIN

Örnekler -Java

❑ Örnek : Java'da dizi kullanarak karakter yığıtı sınıfı oluşturma ve kullanma.

❑ `import java.io.*;`

❑ `class StackChar`

❑ `{`

❑ `private int maxSize; private char[] stackArray; private int top;`

❑ `public StackChar(int max)`

❑ `{ maxSize = max; stackArray = new char[maxSize]; top = -1; }`

❑ `public void push(char j) { stackArray[++top] = j; }`

❑ `public char pop() { return stackArray[top--]; }`

❑ `public boolean isEmpty() { return top == -1; }`

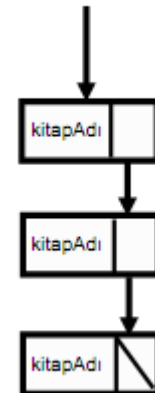
❑ `}`

Örnekler –C#

```
? Örnek: Bağlı liste ile yığıt oluşturma
? class stackNodeC
? {
? public string kitapAdi;    public stackNodeC sonraki;
? public stackNodeC(string kitapAdi) { this.kitapAdi = kitapAdi; }
? }

? class stackC
? {
? public stackNodeC headNode;
? public stackC(string kitapAdi)
? {
?     this.headNode = new stackNodeC(kitapAdi);
?     this.headNode.sonraki = headNode;
? }
? }
? stackC kitapYigin = new stackC("");
?
```

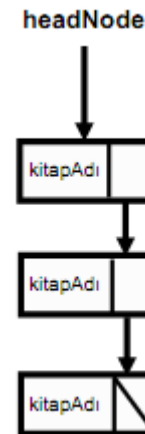
headNode



Örnekler –C#

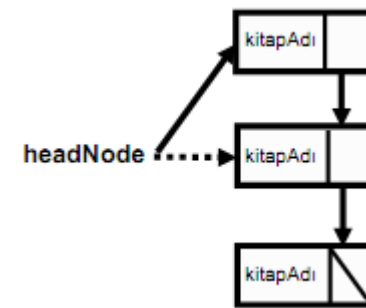
? /* Stack işlemleri :
? boş yığın `stackSize() == 0` eleman sayısı= `stackSize()`
? eleman ekleme= `push(kitapAdi)` eleman alma= `pop()` */

```
? public int stackSize()  
? {  
?     stackNodeC aktif = new stackNodeC("");  
?     aktif = kitapYigin.headNode;  
?     int i = 0;  
?     while (aktif.sonraki != aktif)  
?     {  
?         aktif = aktif.sonraki; i++;  
?     }  
?     return i;  
? }
```



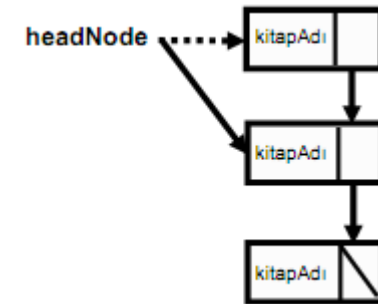
Örnekler –C#

```
? // Stack işlemleri (eleman ekleme)
? public void push(string kitapAdi)
? {
?     if (stackSize() >= MaxSize)
?         MessageBox.Show ("Yığın maksimum elemana sahip ! Yeni
?         eleman eklenemez !", "Dikkat");
?     else
?     { stackNodeC yeniNode = new stackNodeC(tBKitapAdi.Text);
?       yeniNode.sonraki = kitapYigin.headNode;
?       kitapYigin.headNode = yeniNode;
?       IStackSize.Text = "Stack Size =
?       "+Convert.ToString(stackSize());
?     }
? }
```



Örnekler –C#

```
? // Stack işlemleri (eleman alma)
? public void pop()
? {
?     if (stackSize() == 0)
?     {
?         MessageBox.Show("Yığında eleman yok !", "Dikkat");
?     }
?     else
?     {
?         kitapYigin.headNode = kitapYigin.headNode.sonraki;
?     }
? }
```



Örnekler –C++

- ❑ **Örnek:** Yığıt (Stack) kullanarak bağlı liste işlemleri. Girilen cümleyi kelimeleri bozmadan tersten yazdır.
- ❑ **#include <stdio.h>**
- ❑ **#include <string.h>**
- ❑ **#include <conio.h>**
- ❑ **#include <stdlib.h>**
- ❑ **struct kelimeler**
- ❑ **{ char kelime[50];**
- ❑ **struct kelimeler *onceki;**
- ❑ **} *tepe,*yenieleman,*yedek;**

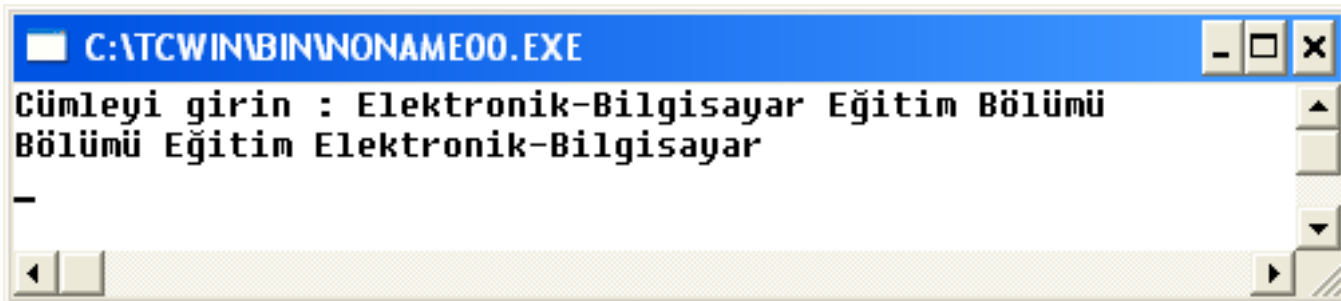
Örnekler –C++

```
? void push(char *gelen) {  
?     yenieleman =(struct kelimeler *) malloc(sizeof(struct kelimeler));  
?     strcpy(yenieleman->kelime, gelen);  
?     yenieleman->onceki = tepe;  
?     tepe = yenieleman; }  
  
? int pop(char *giden) {  
  
?     if (tepe != NULL) {  
?         yedek = tepe;  
?         strcpy(giden, tepe->kelime);  
?         tepe = tepe->onceki;  free(yedek);  return 0;  }  
?     else  
?         return 1; /* yığın boş */
```


Örnekler –C++

```
? main() {  
?   char *yenikelime, *cumle;  
?   tepe = NULL;  
?  
?   printf("Cümleyi girin : "); gets(cumle);  
?   yenikelime = strtok(cumle, " "); // cumle değişkenini boşluğa göre  
?   ayırma.  
?  
?   while (yenikelime) {  
?       push (yenikelime);  
?       yenikelime = strtok(NULL, " ");  
?   }  
?   yenikelime = (char *) malloc(20); /* yenikelime NULL idi, yer açalım*/  
?   while (!pop(yenikelime)) printf("%s ", yenikelime);  
?   printf("\n");  
?   getch();  
? }
```

Örnekler –C++



```
C:\TCWIN\BIN\WONAME00.EXE
Cümleyi girin : Elektronik-Bilgisayar Eğitim Bölümü
Bölümü Eğitim Elektronik-Bilgisayar
_
```

Örnekler –Java

```
? class Reverse
? {
?   public static void main(String args[])
?   {
?       StackChar y = new StackChar(100);
?       String str = "Merhaba";
?       for(int i=0; i<str.length(); ++i)
?           y.push(str.charAt(i));
?       while(! y.isEmpty() ) System.out.println(y.pop());
?   }
? }
```

Yığıt (stack) yapısının dinamik dizi ile gerçekleştirimi

- ❑ Stack.java
- ❑ `public interface Stack<T>`
- ❑ `{ public boolean empty(); // yığıt boş mu test eder`
- ❑ `public T top(); // tepedeki elemanı ver (silme)`
- ❑ `public T pop(); // tepedeki elemanı sil ve ver`
- ❑ `public void push(T item); // item'i yığıtın tepesine ekle`
- ❑ `public void clear(); // Yığıtı boşalt }`

Yığıt (stack) yapısının dinamik dizi ile gerçekleştirimi

```
?   ArrayStack.java
?   public class ArrayStack<T> implements Stack<T>
?   {
?   private static final int DEFAULT_SIZE = 10;
?   private T[] array; // yığıt elemanlarını tutan dizi
?   private int top; // son eklenen elemanın indisi
?   public ArrayStack() // kurucu sınıf
?   { array = (T[]) new Object[DEFAULT_SIZE]; top = -1; }
?   public boolean empty() // yığıt boş mu test eder
?   { return (top== -1); }
```

Yığıt (stack) yapısının dinamik dizi ile gerçekleştirimi

? `ArrayStack.java`

? `public T top() // tepedeki elemanı ver (silme)`

? `{ if (empty()) return null;`

? `return array[top];`

? `}`

? `public T pop() // tepedeki elemanı sil ve ver`

? `{ if (empty()) return null;`

? `return array[top--];`

? `}`

Yığıt (stack) yapısının dinamik dizi ile gerçekleştirimi

```
?   ArrayStack.java
?   public void push(T item)    // item'i yığıtın tepesine ekle
?       { if (top+1==array.length)
?           { T[] newArray = (T[]) new Object[array.length * 2];
?               for (int i=0; i<array.length; i++)
?                   newArray[i] = array[i];
?                   array = newArray;
?               }
?       array[++top] = item;
?   }
?   public void clear()        // Yığıtı boşalt
?   {   top = -1;   }
?   }
```

Yığıt (stack) yapısının dinamik dizi ile gerçekleştirimi

```
? TestArrayStack.java
? public class TestArrayStack
? {
?     public static void main(String[] args)
?     { Stack<String> yigit = new ArrayStack<String>();
?       yigit.push("a");      yigit.push("b");
?       System.out.println("Yigittaki son eleman: " + yigit.top());
?       yigit.push("c");
?       String s = yigit.pop();
?       System.out.println("Yigittan silinen eleman: " + s);
?       yigit.push("d");
?     }
? }
```


Yığıt (stack) yapısının ArrayList ile gerçekleştirimi

```
? Stack.java // Daha önce verildi
? ArrayListStack.java
? import java.util.*;
? public class ArrayListStack<T> implements Stack<T>
? { private ArrayList<T> array; // yığıt elemanlarını tutan dizi
?   public ArrayListStack() // kurucu sınıf
?   { array = new ArrayList<T>(); }
?   public boolean empty() // yığıt boş mu test eder
?   { return array.size()==0; }
?
?   public T top() // tepedeki elemanı ver (silme)
?   { if (empty()) return null;
?     return array.get(array.size()-1);
?   }
? }
```

Yığıt (stack) yapısının ArrayList ile gerçekleştirimi

```
? ArrayListStack.java
? public T pop()          // tepedeki elemanı sil ve ver
? {
?     if (empty()) return null;
?     return array.remove(array.size()-1);
? }
? public void push(T item) // item'i yığıtın tepesine ekle
? {     array.add(item); }
? public void clear()      // Yığıtı boşalt
? {     array.clear(); }
? }
```

Yığıt (stack) yapısının ArrayList ile gerçekleştirimi

```
? TestArrayListStack.java
? public class TestArrayListStack
? {
?     public static void main(String[] args)
?     {
?         Stack<String> yigit = new ArrayStack<String>();
?         yigit.push("a");
?         yigit.push("b");
?         System.out.println("Yigittaki son eleman: " + yigit.top());
?         yigit.push("c");
?         String s = yigit.pop();
?         System.out.println("Yigittan silinen eleman: " + s);
?         yigit.push("d");
?     }
? }
```

Yığıt (stack) yapısının bağlantılı liste ile gerçekleştirimi

```
? Stack.java // Daha önce verildi
? LinkedListStack.java
? public class LinkedListStack<T> implements Stack<T>
? { private Node<T> top = null; // yığıtın tepesindeki eleman
?   public boolean empty() // yığıt boş mu test eder
?   { return top==null; }
?   public T top() // tepedeki elemanı ver (silme)
?   { if (empty()) return null;
?     return top.data;
?   }
?   public T pop() // tepedeki elemanı sil ve ver
?   { if (empty()) return null;
?     T temp = top.data;
?     top = top.next;
?     return temp;
?   }
? }
```

Yığıt (stack) yapısının bağlantılı liste ile gerçekleştirimi

```
? LinkedListStack.java
? public void push(T item) // item'i yığıtın tepesine ekle
?     { Node<T> newNode = new Node<T>();
?       newNode.data = item;
?       newNode.next = top;
?       top = newNode;
?     }
? public void clear() // Yığıtı boşalt
?     { top = null; }
? // Yığıt elemanlarını tutan liste elemanı (inner class)
? class Node<T>
?     { public T data;
?       public Node<T> next;
?     }
? }
```

Yığıt (stack) yapısının bağlantılı liste ile gerçekleştirimi

```
? TestLinkedListStack.java
? public class TestLinkedListStack
? {   public static void main(String[] args)
?     {   Stack<Character> yigit = new LinkedListStack<Character>();
?         yigit.push('a');
?         yigit.push('b');
?         yigit.push('c');
?         System.out.println("Yigittaki elemanlar cikariliyor: ");
?         while (!yigit.empty())
?         {
?             System.out.println(yigit.pop());
?         }
?     }
? }
```

Hanoi Kuleleri– Özyinelemeli Çözüm- C#

```
? using System;  
?  
? using System.Collections.Generic;  
?  
? using System.Text;  
?  
? namespace Hanoi  
?  
? { class Program  
?  
?     { static void Main(string[] args)  
?         { int x;  char from='A', to='B', help='C';  
?         do {  
?  
?             try  
?  
?                 { Console.Write(" input number of disk: "); x = Int32.Parse(Console.ReadLine()); }  
?  
?                 catch (FormatException e) { x = -10; }  
?  
?             } while(x== -10 || x>10);  
?  
?         }  
?  
?     }  
?  
? }
```

Hanoi Kuleleri– Özyinelemeli Çözüm- C#

```
? Console.WriteLine("\n from = A, to = B, help = C\n");  
?  
? hanoi(x, from, to, help);    Console.Read();  
?  
? }  
?  
? static void hanoi(int x, char from, char to, char help)  
? {   if (x > 0)  
?     {   hanoi(x - 1, from, help, to);  
?         move(x, from, to);  
?         hanoi(x - 1, help, to, from);  
?     }  
? }  
?  
? static void move(int x, char from, char to)  
? { Console.WriteLine(" move disk "+x+" from "+from+" to "+to); }  
?  
? }  
?
```