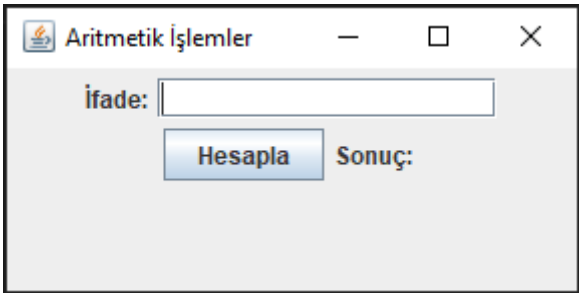
Programlama Dilleri Ödevi



BNF tanımımız:

<expression> ::= <term> | <expression> "+" <term>

<term> ::= <factor> | <term> "*" <factor>

<factor> ::= "(" <expression> ")" | <number>

<number> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9

Ödev için açıklamalar

Kodumuz 2 sınıftan oluşmaktadır:

1. **Kullanıcı Arayüzü (GUI)** - AritmetikAyrıştirici sınıfı
2. **Parser** - Ayrıştirici sınıfı

Bu sınıfların açıklamaları aşağıdadır.

1-AritmetikAyrıştirici Sınıfı

Bu sınıf bize jframe sınıfını kullanarak uygulamamız için arayüz sağlamaktadır.

Sınıf kullanıcıdan bir aritmetik ifade almamızı ve hesapla butonuna basıldığında ifadeyi parse ederek sonucu ekranda gösteren bir etkileşim sağlar.

Kullandığımız yapılar:

* JTextField inputarea: Kullanıcının ifade girmesi için bir metin kutusu.

* JButton hesapButton: İfadeyi parse edip sonucu hesaplamak için buton. üzerine tıklandığında, inputarea içindeki metin alınır ve Ayrıştirici sınıfına gönderilir.

* JLabel resultarea: Hesaplanan sonucu veya hata mesajını göstermek için bir label. Parse işlemi başarılıysa sonuç resultarea etiketine yazdırılır. Hatalı ifade girilirse "Geçersiz ifade!" mesajı görüntülenir.

2- Ayrıştirici Sınıfı

Bu sınıf kullanıcının girdiği ifadeyi BNF kurallarına göre parse eden ve işlemleri gerçekleştiren parser sınıfıdır.

Sınıf içindeki değişkenlerimiz:

- String input: Kullanıcının girdiği ifade (boşluk olmadan).
- int pozisyon: Girdi ifadesindeki mevcut konumu takip eden indeks.

2.1 ifadeAyrıştir Metodu

- BNF'deki <expression> kuralına göre ifadeyi çözümler.
- İlk olarak terimAyrıştir metodunu çağırır ve toplama işlemlerini (+ sembolü) ele alır.

2.2 terimAyrıştir Metodu

- BNF'deki <term> kuralına göre çalışır.
- faktorAyrıştir metodunu çağırarak çarpma işlemlerini (* sembolü) çözümler.

2.3 faktorAyrıştir Metodu

- <factor> kuralını parse eder. Bu kuralda iki ihtimal vardır:
    1. Parantez içine alınmış bir ifade (örneğin (2+3)*4).
    2. Sadece bir sayı (<number>).
- Parantezli ifadeleri çözümlemek için ifadeAyrıştir metodunu yeniden çağırır.

2.4 sayiAyrıştir Metodu

- <number> kuralına göre çalışır ve sayı karakterlerini çözümler.
- Sayı olmayan bir karakterle karşılaşırsa hata verir.

KODLAR SONUÇ KISMININ ALTINDA VERİLMİŞTİR.

Sonuç:

Kullanıcı arayüze bir ifade girip "Hesapla" butonuna tıkladığında inputarea'den alınan metin beyaz boşluklardan arındırılır. Ayrıştirici sınıfındaki metodlar sırasıyla çağrılarak ifade BNF kurallarına göre parse edilir. Parse işlemi sonucunda hesaplanan değer resultarea etiketine yazdırılır. Eğer geçersiz bir ifade varsa, hata mesajı gösterilir.

Kodlar:

```
import javax.swing.*;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;


public class AritmetikAyrişţirici extends JFrame {

  private JTextField inputarea;

  private JLabel resultarea;


  public AritmetikAyrişţirici() {

    setTitle("Aritmetik İşlemler ");

    setSize(300, 150);

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    setLayout(new FlowLayout());


    inputarea = new JTextField(15);

    JButton hesapButton = new JButton("Hesapla");

    resultarea = new JLabel("Sonuç: ");


    // Butona tiklama işlemi

    hesapButton.addActionListener(new ActionListener() {

      @Override
```

```
        public void actionPerformed(ActionEvent e) {
            try {
                String input = inputarea.getText();
                Ayrıştirici ayrıştirici = new Ayrıştirici(input);
                int result = ayrıştirici.ifadeAyrıştir();
                resultarea.setText("Sonuç: " + result);
            } catch (Exception ex) {
                resultarea.setText("Geçersiz ifade!");
            }
        }
    });

    // Bileşenleri pencereye ekleme
    add(new JLabel("İfade:"));
    add(inputarea);
    add(hesapButton);
    add(resultarea);
  }

  public static void main(String[] args) {
    AritmetikAyrıştirici gui = new AritmetikAyrıştirici();
    gui.setVisible(true);
    ;
  }
}

// Ayrıştirici sinifi
class Ayrıştirici {
```

```
private String input;
private int loc;

public Ayrıştirici(String input) {
   this.input = input.replaceAll("\\s+", ""); // Beyaz boşluklari kaldir
   this.loc = 0;
}

public int ifadeAyrıştir() {
   int result = terimAyrıştir();
   while (loc < input.length() && input.charAt(loc) == '+') {
      loc++;
      result += terimAyrıştir();
   }
   return result;
}

private int terimAyrıştir() {
   int result = faktorAyrıştir();
   while (loc < input.length() && input.charAt(loc) == '*') {
      loc++;
      result *= faktorAyrıştir();
   }
   return result;
}

private int faktorAyrıştir() {
   if (loc < input.length() && input.charAt(loc) == '(') {
```

```
            loc++;
            int result = ifadeAyrıştir();
            if (loc < input.length() && input.charAt(loc) == ')') {
                loc++;
            } else {
                throw new RuntimeException("Kapatma parantezi eksik!");
            }
            return result;
        } else {
            return sayiAyrıştir();
        }
    }

    private int sayiAyrıştir() {
        StringBuilder sayi = new StringBuilder();
        while (loc < input.length() && Character.isDigit(input.charAt(loc))) {
            sayi.append(input.charAt(loc++));
        }
        if (sayi.length() == 0) {
            throw new RuntimeException("Sayi bekleniyor!");
        }
        return Integer.parseInt(sayi.toString());
    }
}
```

Sueda Onur

230260145