KARADENİZ TEKNİK ÜNİVERSİTESİ

SİMGESEL MATEMATİK VE PROGRAMLAMA

2.Arasınav Ödevi Cevapları

Bilgisayar Mühendisliği Doktora Öğrencisi

HAKAN AYDIN

379449

KODLAR

1. Expression.jj Kodları

Bu alt başlık Expression.jj dosyasındaki kodları içermektedir. Programın büyük bir kısmını oluşturan parser çözüme ulaşmada yol gösterici olarak rol almaktadır. Vizeden farklı olarak burada parantezlerin de tanımlaması yapılmaktadır. “A” ifadesi parantez içindeki ifadeleri ve işlemleri içermektedir. Burada E tanımlaması A’nın numa gitmesini ya da E’nin kendini çağırmasını sağlamaktadır.

```
options {
  DEBUG_PARSER = false;
}

PARSER_BEGIN(ExpressionParser)
 public class ExpressionParser {
  public static void main(String[] args) {
              try {
              Exp exp = new ExpressionParser(System.in).Parse();
     EvalVisitor evalVisitor = new EvalVisitor();
     System.out.println(exp.accept(evalVisitor));
              }
              catch(ParseException ex) {
              System.out.println("Fail!\n" + ex.getMessage());
              }
  }
}
PARSER_END(ExpressionParser)

TOKEN: {
   <PLUS: "+">
 | <MINUS: "-">
 | <TIMES: "*">
 | <DIVIDE: "/">
 | <RPR: ")">
 | <LPR: "(">
 | <NUM: (["0"-"9"])+ >
}

SKIP: { " " | "\t" | "\r" | "\r\n" | "\n"}

/*
E - > A

A -> T ((+|-) T)*

T -> (F|D) ((*|/) F)*

F - > num

D - > "(" E ")"
*/

Exp Parse():
{ Exp a;}
{
              a = E() (<EOF>)
  { return a; }
}

Exp E():
```

```
{ Exp a, b; Token t;}
{
    (a = A()  )
  { return a; }
}

Exp A():
{ Exp a, b; int k = 1; }
{
   ( <PLUS> | <MINUS> { k = -1; })?
    a=T() { a = k > 0 ? a : new Times(a, new Num(-1+"")); }
    (
     <PLUS>  b=T() { a=new Plus(a, b);  }
    | <MINUS>  b=T() { a=new Minus(a, b); }
    )*
    { return a; }
}

Exp T() :
{ Exp a, b;}
{
   (a=F()|a = D())
 (
   <TIMES> (b=F()|b = D()) { a=new Times(a, b);  }
 | <DIVIDE> (b=F()|b = D()) { a=new Divide(a, b); }
 )*
 { return a; }
}

Exp F():
{ Exp a, b; Token t;}
{
            t=<NUM> { a = new Num(t.image); }
   //|  a = D()
    // b =  E() { a = new Exp(a,b); }
            { return a; }
}

Exp D():
{ Exp a, b; Token t;}
{
   <LPR> a = E() <RPR>
   { return a; }
}
```

2. Visitor.java Kodları

Bu kısım Visitor.java dosyasına ait kodları içermektedir.

```
interface Visitor {
    public String visit(Exp exp);
    public String visit(Plus exp);
    public String visit(Minus exp);
    public String visit(Times exp);
    public String visit(Divide exp);
    public String visit(Num exp);
}
```

3. AST.java Kodları

Bu kısımda AST.java dosyasında bulunan kodlar gösterilmektedir.

```
abstract class Exp {
	public abstract String accept(Visitor v);
}

class Plus extends Exp {
	public  Exp a, b;
	public Plus(Exp x, Exp y) {
	a = x;
	b = y;
	}

	public String accept(Visitor v)
	{
	return v.visit(this);
	}
}

class Minus extends Exp {
	public  Exp a, b;
	public Minus(Exp x, Exp y) {
	a = x;
	b = y;
	}

	public String accept(Visitor v)
	{
	return v.visit(this);
	}
}

class Times extends Exp {
	public  Exp a, b;
	public Times(Exp x, Exp y) {
	a = x;
	b = y;
	}

	public String accept(Visitor v)
	{
	return v.visit(this);
	}
}

class Divide extends Exp {
	public  Exp a, b;
	public Divide(Exp x, Exp y) {
	a = x;
	b = y;
	}

	public String accept(Visitor v)
	{
```

```
            return v.visit(this);
            }
}

class Num extends Exp {
            public  String a;
            public Num(String x) {
            a = x;
            }
            public String accept(Visitor v)
            {
            return v.visit(this);
            }
}
```

4. EvalVisitor.java Kodları

Burada input dosyasından okunan değerler parser aracılığıyla alt elemanlara bölünmektedir. boolean değerleri gelen string ifadesini kontrol ederek exp ifadesinden yeni ifadeleri kontrol etmektedir. Her fonksiyon (minus, times, sum, divide vs.,) için yapılan kontrol ifadesi sadelestirme fonksiyonu ile son şeklini almaktadır. Bu fonksiyonun asıl amacı gelen matematiksel ifadelerin arasına "/" operatörü koyup matematiksel sadeleştirme yapmaktır. Ödev ilk yapıldığında direkt olarak matematiksel bir sonuç üretmekteydi. Fakat biraz uğraşlar sonucunda java tarafında bu sorun çözülerek araya bölme operatörü eklenmiştir.

```
import java.util.Arrays;
import java.util.List;

public class EvalVisitor implements Visitor
{

            public String visit(Exp exp)
            {
             return exp.accept(this);
            }

            public String visit(Plus exp)
   {
             int sayi1 = 1, sayi2=1, sayi3=1, sayi4=1;

             String a = exp.a.accept(this);
             String b = exp.b.accept(this);

            boolean aDurum = (a.contains("/") || a.contains("*") || a.contains("-") || a.contains("+"));
            boolean bDurum = (b.contains("/") || b.contains("*") || b.contains("-") || b.contains("+"));

            a = a.replaceAll("[^-?0-9]+", " ");
             List<String> aDakiSayilar = Arrays.asList(a.trim().split(" "));

             b = b.replaceAll("[^-?0-9]+", " ");
             List<String> bDekiSayilar = Arrays.asList(b.trim().split(" "));

             if (aDurum == true && bDurum == true){//Bu if'e girerse 1/2 + 3/6 gibi bir ifade var demektir
```

```
                    sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//1/2'den 1 alindi.
                    sayi2 = Integer.valueOf(aDakiSayilar.get(1)+"");//1/2'den 2 alindi.

                    sayi3 = Integer.valueOf(bDekiSayilar.get(0)+"");//3/6'den 3 alindi.
                    sayi4 = Integer.valueOf(bDekiSayilar.get(1)+"");//3/6'den 6 alindi.

                    sayi1 = (sayi1 * sayi4) + (sayi3 * sayi2); //yeni pay oluyor.
                    sayi2 = sayi2 * sayi4; //yeni payda oluyor.

            }

            else if (aDurum == false && bDurum == false) {//Bu if'e girerse 2 + 3 gibi bir ifade var
demektir
                    sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//2 alindi.
                    sayi2 = Integer.valueOf(bDekiSayilar.get(0)+"");//3 alindi.

                    sayi1 = sayi1 + sayi2; //yeni pay oluyor.
                    sayi2 = 1; //yeni payda oluyor.
            }

            else {
                    if (aDurum == true) {//Bu if'e girerse 1/2 + 3 gibi bir ifade var demektir
                            sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//1/2'den 1 alindi.
                            sayi2 = Integer.valueOf(aDakiSayilar.get(1)+"");//1/2'den 2 alindi.

                            sayi3 = Integer.valueOf(bDekiSayilar.get(0)+"");//3 alindi.

                            sayi1 = (sayi1) + (sayi3 * sayi2); //yeni pay oluyor.
                    }
                    else if(bDurum == true) {//Bu if'e girerse  3 + 1/2 gibi bir ifade var demektir
                            sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//3 alindi.

                            sayi3 = Integer.valueOf(bDekiSayilar.get(0)+"");//1/2'den 1 alindi.
                            sayi4 = Integer.valueOf(bDekiSayilar.get(1)+"");//1/2'den 2 alindi.

                            sayi1 = (sayi1 * sayi4) + (sayi3); //yeni pay oluyor.
                            sayi2 = sayi4; //yeni payda oluyor.
                    }

            }

            int[] sayilar = new int[2];
            sayilar = sadelestirme(sayi1, sayi2);

            return sayilar[0] + "/" + sayilar[1];
    }

    public String visit(Minus exp)
    {
            int sayi1 = 1, sayi2=1, sayi3=1, sayi4=1;

            String a = exp.a.accept(this);
                    String b = exp.b.accept(this);

                    boolean aDurum = (a.contains("/") || a.contains("*") || a.contains("-") ||
a.contains("+"));
                    boolean bDurum = (b.contains("/") || b.contains("*") || b.contains("-") ||
```

```
b.contains("+"));

                a = a.replaceAll("[^-?0-9]+", " ");
        List<String> aDakiSayilar = Arrays.asList(a.trim().split(" "));

        b = b.replaceAll("[^-?0-9]+", " ");
        List<String> bDekiSayilar = Arrays.asList(b.trim().split(" "));

        if (aDurum == true && bDurum == true){//Bu if'e girerse 1/2 + 3/6 gibi bir ifade var demektir

                sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//1/2'den 1 alindi.
                sayi2 = Integer.valueOf(aDakiSayilar.get(1)+"");//1/2'den 2 alindi.


                sayi3 = Integer.valueOf(bDekiSayilar.get(0)+"");//3/6'den 3 alindi.
                sayi4 = Integer.valueOf(bDekiSayilar.get(1)+"");//3/6'den 6 alindi.

                sayi1 = (sayi1 * sayi4) - (sayi3 * sayi2); //yeni pay oluyor.
                sayi2 = sayi2 * sayi4; //yeni payda oluyor.

        }

        else if (aDurum == false && bDurum == false) {//Bu if'e girerse 2 + 3 gibi bir ifade var
demektir
                sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//2 alindi.
                sayi2 = Integer.valueOf(bDekiSayilar.get(0)+"");//3 alindi.

                sayi1 = sayi1 - sayi2; //yeni pay oluyor.
                sayi2 = 1; //yeni payda oluyor.
        }

        else {
                if (aDurum == true) {//Bu if'e girerse 1/2 + 3 gibi bir ifade var demektir
                        sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//1/2'den 1 alindi.
                        sayi2 = Integer.valueOf(aDakiSayilar.get(1)+"");//1/2'den 2 alindi.

                        sayi3 = Integer.valueOf(bDekiSayilar.get(0)+"");//3 alindi.

                        sayi1 = (sayi1) - (sayi3 * sayi2); //yeni pay oluyor.
                }
                else if(bDurum == true) {//Bu if'e girerse  3 + 1/2 gibi bir ifade var demektir
                        sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//3 alindi.

                        sayi3 = Integer.valueOf(bDekiSayilar.get(0)+"");//1/2'den 1 alindi.
                        sayi4 = Integer.valueOf(bDekiSayilar.get(1)+"");//1/2'den 2 alindi.

                        sayi1 = (sayi1 * sayi4) - (sayi3); //yeni pay oluyor.
                        sayi2 = sayi4; //yeni payda oluyor.
                }

        }

        int[] sayilar = new int[2];
        sayilar = sadelestirme(sayi1, sayi2);

        return sayilar[0] + "/" + sayilar[1];
   }
```

```
   public String visit(Times exp)
   {
          int sayi1 = 1, sayi2=1, sayi3=1, sayi4=1;

          String a = exp.a.accept(this);
                String b = exp.b.accept(this);

                boolean aDurum = (a.contains("/") || a.contains("*") || a.contains("-") ||
a.contains("+"));
                boolean bDurum = (b.contains("/") || b.contains("*") || b.contains("-") ||
b.contains("+"));

                a = a.replaceAll("[^-?0-9]+", " ");
          List<String> aDakiSayilar = Arrays.asList(a.trim().split(" "));

          b = b.replaceAll("[^-?0-9]+", " ");
          List<String> bDekiSayilar = Arrays.asList(b.trim().split(" "));

          if (aDurum == true && bDurum == true){//Bu if'e girerse 1/2 * 3/6 gibi bir ifade var demektir
                sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//1/2'den 1 alindi.
                sayi2 = Integer.valueOf(aDakiSayilar.get(1)+"");//1/2'den 2 alindi.

                sayi3 = Integer.valueOf(bDekiSayilar.get(0)+"");//3/6'den 3 alindi.
                sayi4 = Integer.valueOf(bDekiSayilar.get(1)+"");//3/6'den 6 alindi.

                sayi1 = sayi1 * sayi3; //yeni pay oluyor.
                sayi2 = sayi2 * sayi4; //yeni payda oluyor.

          }

          else if (aDurum == false && bDurum == false) {//Bu if'e girerse 2 * 3 gibi bir ifade var
demektir
                sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//2 alindi.
                sayi2 = Integer.valueOf(bDekiSayilar.get(0));//3 alindi.

                sayi1 = sayi1 * sayi2; //yeni pay oluyor.
                sayi2 = 1; //yeni payda oluyor.
          }

          else {
                if (aDurum == true) {//Bu if'e girerse 1/2 * 3 gibi bir ifade var demektir
                      sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//1/2'den 1 alindi.
                      sayi2 = Integer.valueOf(aDakiSayilar.get(1)+"");//1/2'den 2 alindi.

                      sayi3 = Integer.valueOf(bDekiSayilar.get(0));//3 alindi.

                      sayi1 = sayi1 * sayi3; //yeni pay oluyor.
                }
                else if(bDurum == true) {//Bu if'e girerse  3 * 1/2 gibi bir ifade var demektir
                      sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//3 alindi.

                      sayi3 = Integer.valueOf(bDekiSayilar.get(0)+"");//1/2'den 1 alindi.
                      sayi4 = Integer.valueOf(bDekiSayilar.get(1)+"");//1/2'den 2 alindi.

                      sayi1 = sayi1 * sayi3; //yeni pay oluyor.
                      sayi2 = sayi4; //yeni payda oluyor.
```

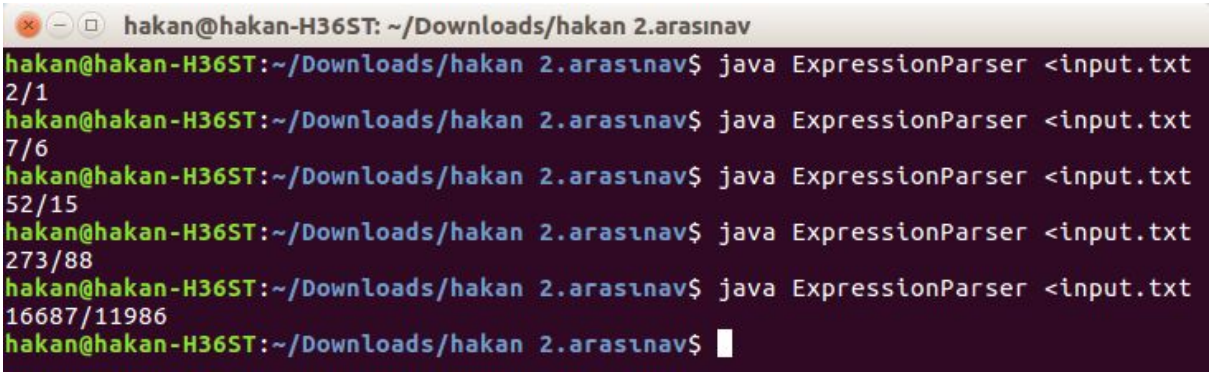
```
                }

        }

        int[] sayilar = new int[2];
        sayilar = sadelestirme(sayi1, sayi2);

        return sayilar[0] + "/" + sayilar[1];
    }

    public String visit(Divide exp)
    {
        int sayi1 = 1, sayi2=1, sayi3=1, sayi4=1;

        String a = exp.a.accept(this);
        String b = exp.b.accept(this);

        boolean aDurum = (a.contains("/") || a.contains("*") || a.contains("-") || a.contains("+"));
        boolean bDurum = (b.contains("/") || b.contains("*") || b.contains("-") || b.contains("+"));

        a = a.replaceAll("[^-?0-9]+", " ");
        List<String> aDakiSayilar = Arrays.asList(a.trim().split(" "));

        b = b.replaceAll("[^-?0-9]+", " ");
        List<String> bDekiSayilar = Arrays.asList(b.trim().split(" "));

        if (aDurum == true && bDurum == true){//Bu if'e girerse 1/2 / 3/6 gibi bir ifade var demektir
                sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//1/2'den 1 alindi.
                sayi2 = Integer.valueOf(aDakiSayilar.get(1)+"");//1/2'den 2 alindi.

                sayi3 = Integer.valueOf(bDekiSayilar.get(0)+"");//3/6'den 3 alindi.
                sayi4 = Integer.valueOf(bDekiSayilar.get(1)+"");//3/6'den 6 alindi.

                sayi1 = sayi1 * sayi4; //yeni pay oluyor.
                sayi2 = sayi2 * sayi3; //yeni payda oluyor.

        }

        else if (aDurum == false && bDurum == false) {//Bu if'e girerse 2 / 3 gibi bir ifade var
demektir
                sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//2 alindi.
                sayi2 = Integer.valueOf(bDekiSayilar.get(0)+"");//3 alindi.

                sayi1 = sayi1; //yeni pay oluyor.
                sayi2 = sayi2; //yeni payda oluyor.
        }

        else {
                if (aDurum == true) {//Bu if'e girerse 1/2 / 3 gibi bir ifade var demektir
                        sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//1/2'den 1 alindi.
                        sayi2 = Integer.valueOf(aDakiSayilar.get(1)+"");//1/2'den 2 alindi.

                        sayi3 = Integer.valueOf(bDekiSayilar.get(0)+"");//3 alindi.

                        sayi2 = sayi2 * sayi3; //yeni pay oluyor.
                }
                else if(bDurum == true) {//Bu if'e girerse  3 / 1/2 gibi bir ifade var demektir
```

```
                        sayi1 = Integer.valueOf(aDakiSayilar.get(0)+"");//3 alindi.

                        sayi3 = Integer.valueOf(bDekiSayilar.get(0)+"");//1/2'den 1 alindi.
                        sayi4 = Integer.valueOf(bDekiSayilar.get(1)+"");//1/2'den 2 alindi.

                        sayi1 = sayi1 * sayi4; //yeni pay oluyor.
                        sayi2 = sayi3; //yeni payda oluyor.
                }

        }

        int[] sayilar = new int[2];
        sayilar = sadelestirme(sayi1, sayi2);

        return sayilar[0] + "/" + sayilar[1];
  }

        public String visit(Num exp)
        {
        return exp.a + "/" + 1;
        }

        public int[] sadelestirme(int sayi1, int sayi2) {
         int kucuk = sayi1;
         if (kucuk  > sayi2) {
                kucuk = sayi2;
         }
         for(int i=1; i<=kucuk;i++) {
                if((sayi1%i==0) && (sayi2%i==0)) {
                        sayi1 /= i;
                        sayi2 /= i;
                }
         }
         int[] sayilar = new int[2];
         sayilar[0] = sayi1;
         sayilar[1] = sayi2;

         return sayilar;
        }
}
```

Ekran çıktısı

Aşağıda verilen input verilerine göre ekran çıktıları gösterilmektedir. input.txt dosyasına sırası ile aşağıdaki ifadeler veriler sonuçlar elde edilmiştir.

Aritmetik ifade (input data)
-1+3
(1/2+3)*2/6
2*(3-2/5)/(1+1/2)
1+(2+3/4)/2+2/(2+3/4)
1+1/(2+2/(3+3/(4+4/(5+5/(6+6/(7))))))


hakan@hakan-H36ST: ~/Downloads/hakan 2.arasınav
hakan@hakan-H36ST:~/Downloads/hakan 2.arasınav$ java ExpressionParser <input.txt
2/1
hakan@hakan-H36ST:~/Downloads/hakan 2.arasınav$ java ExpressionParser <input.txt
7/6
hakan@hakan-H36ST:~/Downloads/hakan 2.arasınav$ java ExpressionParser <input.txt
52/15
hakan@hakan-H36ST:~/Downloads/hakan 2.arasınav$ java ExpressionParser <input.txt
273/88
hakan@hakan-H36ST:~/Downloads/hakan 2.arasınav$ java ExpressionParser <input.txt
16687/11986
hakan@hakan-H36ST:~/Downloads/hakan 2.arasınav$