

Vorlesung Grundlagen der Programmiersprachen Sommer 2004 - Lösung zu Blatt 6

Lösung zu Aufgabe 16

a)

```
Kommentar = Kommentare | {kk}           // Kommentar oder leer
  Vok = Terminal | Nichtterminal        // Terminal oder Nichtterminal
  Prod = Vok list x Kommentar           // Liste mit Elementen aus Vok und ein Kommentar
  Gram = Kommentar x Prod list          // Ein Kommentar und eine Liste von Produktionen
```

b) Programmtext für Aufgabenteile b) und c):

```
/homes/info-f/gdp/aufgaben/blatt6/blatt6_sol/a16.java)
```

c) Programmtext für Aufgabenteile b) und c):

```
/homes/info-f/gdp/aufgaben/blatt6/blatt6_sol/a16.java)
```

Lösung zu Aufgabe 17

Typ-Definition der Klasse "Paar<A,B>":

```
Paar = 'a X 'b
```

Das Pizza-System erzeugt zwei Java-Dateien:

Paar.java

```
class Paar {
    private Object erster;
    private Object zweiter;
    public Paar(Object erster, Object zweiter) {
        super();
        this.erster = erster;
        this.zweiter = zweiter;
    }
    public Object Paar$getErster() {
        return erster;
    }
    public Object Paar$getZweiter() {
        return zweiter;
    }
    public String toString() {
        return "(" + erster + ", " + zweiter + ")";
    }
}
```

und PizzaTest.java

```
class PizzaTest {
    public static void main(String[] argv) {
        Paar nummern = new Paar((Object)new Integer(17), (Object)new Integer(01754321));
        Paar telefon = new Paar((Object)"Schmidt", nummern.Paar$getZweiter());
        System.out.println(nummern + "/" + telefon);
    }
    PizzaTest() {
        super();
    }
}
```

Beobachtungen

- Die Auftreten der generischen Typparameter werden durch den Java-Typ `Object` ersetzt.
- Methoden mit generischem Ergebnistyp werden in `KlassenName$Methodenname` umbenannt.
- Aktuelle Parameter zu generischen Typen
 - werden an `Object` angepasst.
 - Werte von Grundtypen (z.B. `int`) werden durch `Wrapper`-Objekte ersetzt.
- Konstruktoren erhalten explizite `super ()`-Aufrufe.

Lösung zu Aufgabe 18

Das folgende MYSTERY Programm erzeugt zwei Ausgaben. Aus den Ausgaben und der Initialisierung von `global := 1` kann man ablesen, dass MYSTERY Parameter mittels "call-by-value-and-result" übergibt.

```
VAR global:INTEGER;

PROCEDURE assign(p:INTEGER):INTEGER=
BEGIN
  p := p + 1;
  PRINT global;
  RETURN p;
END;

PROCEDURE main():INTEGER=
VAR void : INTEGER;
BEGIN
  global := 1;
  void := assign(global);
  PRINT global;
  RETURN global;
END;
```

Ausgabe des obigen Programms:

```
User output: 1
User output: 2
```

Begründung:

Da sich der Wert von `global` durch Ausführung von `assign` verändert, scheidet `call-by-value` aus der Kandidatenliste der Parameterübergabearten aus.

Da während der Ausführung von `assign` die Modifikation des formalen Parameter `p` nicht auf `global` durchschlägt, scheidet auch `call-by-reference` aus der Kandidatenliste aus.