

a) Ableitung des ersten Java-Programms:

```

<Klasse>
class <Bezeichner> { <Variablenliste> <Methodenliste> }
class <Buchstabe> <Zeichenkette> { <Variablenliste> <Methodenliste> }
class M <Zeichenkette> { <Variablenliste> <Methodenliste> }
class M <Zeichen> <Zeichenkette> { <Variablenliste> <Methodenliste> }
class M <Buchstabe> <Zeichenkette> { <Variablenliste> <Methodenliste> }
class Mo <Zeichenkette> { <Variablenliste> <Methodenliste> }
...
class Mover { <Variablenliste> <Methodenliste> }
class Mover { <Variable> ; <Variablenliste> <Methodenliste>}
class Mover { <Variable> ; <Variable> ; <Variablenliste> <Methodenliste>}
class Mover { <Variable> ; <Variable> ; <Methodenliste>}
class Mover { <Datentyp> <Bezeichner> ; <Variable> ; <Methodenliste>}
class Mover { int <Bezeichner> ; <Variable> ; <Methodenliste>}
class Mover { int x; <Variable> ; <Methodenliste>}
class Mover { int x; int y ; <Methodenliste>}
class Mover { int x; int y ; <Methode> <Methodenliste> }
class Mover { int x; int y ; <Methode> }
class Mover { int x; int y ; void <Bezeichner> () { <Variablenliste> <Befehlsliste> } }
class Mover { int x; int y ; void move() { <Variablenliste> <Befehlsliste> } }
class Mover { int x; int y ; void move() { <Befehlsliste> } }
class Mover { int x; int y ; void move() { <Befehl> <Befehlsliste> } }
class Mover { int x; int y ; void move() { <Befehl> <Befehl> <Befehlsliste> } }
class Mover { int x; int y ; void move() { <Befehl> <Befehl> } }
class Mover { int x; int y ; void move() { <Bezeichner> = <Wert> ; <Bezeichner> = <Wert> ; } }
class Mover { int x; int y ; void move() { x = 10 ; y = 20 ; } }

```

Ableitung des zweiten Programms:

```
<Klasse>  
class <Bezeichner> { <Variablenliste> <Methodenliste> }  
class <Buchstabe> <Zeichenkette> { <Variablenliste> <Methodenliste> }
```

Da <Buchstabe> sich nicht durch die Ziffer 1 ersetzen lässt, kann das gegebene Programm aus dieser Grammatik nicht erzeugt werden. Das entspricht tatsächlich auch den Regeln der „echten“ Programmiersprache Java: Ein Bezeichner darf nicht mit einer Ziffer beginnen. Auch für die Zeilen `int x, y;` oder `x = x + 10;` gibt es keine Regeln in der vorgegebenen Grammatik.

b) Ergänzende Regeln für Rechnungen:

```
<Befehl>    →    <Bezeichner> = <Operand> <Operator> <Operand> ;  
<Operand>  →    <Bezeichner> | <Zahl>  
<Operator> →    + | - | * | /
```