

1**Den Datentypen gültig oder ungültig zuweisen mit Begründung**

int a, double b, short c, byte d, long e, float f, char g, boolean h, string i;

- a) (a<++b) **UNGÜLTIG** <++ ist kein Statement (?)
- b) i.charAt(b) **UNGÜLTIG** charAt nur bei int möglich
- c) ~a **GÜLTIG** binäres Komplement bei int erlaubt
- d) f%f **GÜLTIG** float und double sind gültige modulo Werte
- e) i+i **GÜLTIG** 2 Strings kann man addieren
- f) a>>3 **GÜLTIG** rechtsshift ist bei 32Bit des Integer-Typs erlaubt

2**Den/Die Fehler finden im folgenden Code**

```

a)
import eprog.*;
class Mult
{
static void main (String argv[])
{
int n=0;
int m=0;
try {
    m=EprogIO.readInt();
    n=EprogIO.readInt();
}
catch (Exception e)
{
    EprogIO.println("?");
}
    EprogIO.println(b(m,n));
}
static int b (int a, int b)
{
    return (b==0)?0: (b==1)?a: (b==2)?b(a<<1,--b);a(a,b(a,--b));
}
static int a (int a, int b)
{
    return (a==1)?++b:a(a--,++b);
}
}

```

```

.
.
.
static int b (int a, int b)
{
    return (b==0)?0: (b==1)?a: (b==2)?b(a<<1,--b):a(a,b(a,--b));
}
static int a (int a, int b)
{
    return (a==1)?++b:a(--a,++b);
}

```

b) Ein netter kleiner Quellcode ist gegeben (2 A4 Seiten) und man muss die Fehler anstreichen (Klammer und PunktStrich Fehler).

3

Methoden erstellen

Klasse Matrix schreiben, die Klasse erbt und Erweiterungen hat.

```
public simpleMatrix (int rows, int columns)
```

```
public int getRows()
```

```
public int getColumns()
```

```
public void setElement (int row, int column, int value) throws
OutOfMatrixDimension Exception
```

```
public int getElement (int row, int column) throws OutOfMatrix
Exception
```

```
public void add(simpleMatrix other Matrix)throws
MismatchMatrixDimensionException
```

```
public void print()
```

Zusätzlich noch a) Multiplikation: Aufruf A.multiply(B) → A=A*B

b) Transponieren: A.transpone()

Gab dann noch paar Formeln zum multiplizieren und transponieren

4

Nachvollziehen von einem Programm

Variablen a,b,c sind die letzten 3 Stellen der Matrikelnummer

```
import eprog.*;
public class nachvollziehen
{
    final static int Y=1;
    final static int Z=4;
        static int a=0,b=0;
    public static int A(int x,int y)
    {
        int c=0,b=0;
        {
            int z=5;
            c=z;
            b=y;
        }
        return a;
    }
    public static int B(int x, int y)
    {
        int c=2,a=0;
        {
            int z=c;
            x=b;
        }
        c=x;
        a=y;
        return x;
    }
    public static int C(int x, int y)
    {
        int c=b;
        a=++x+y;
        b=++x;
```

```

    return c;
}
public static void D(int x, int y, int z)
{
    x=++a;
    b+=++y+z;
    z=x;
    y=C(x,x);
}
public static void E(int x, int y, int z)
{
    x=y++;
    z=y++;
    y+=a+++b;
}
public static void F(int x, int y, int z)
{
    a=++x+z++;
}
public static int G(int x, int y)
{
    int a=0;
    a+=x;
    b=y;
    return y;
}
public static void main (String [] args)
{
    int c=0,d=0;
    try {
        a=EprogIO.readInt();
        b=EprogIO.readInt();
        c=EprogIO.readInt();
    }
    catch (EprogException e)
    {
    }
    for (d=Y;d<Z;EprogIO.println(d++))

```

```

{
switch(d)
{
case 0: a=A(b,c);break;
case 1: b=B(a,c);break;
case 2: c=C(c,b);break;
case 3: D(a,b,c);break;
case 4: E(b,a,c);break;
case 5: F(b,c,a);break;
case 6: G(a,c);break;
}
EprogIO.print(a+"\t"+b+"\t"+c+"\t");
}
}
}

```

5

Klassendesign (double)

a) Von einem Teleskop ist 1) Objektivdurchmesser 2) Brennweite Teleskop 3) Brennweite Okular wichtig.

Berechne: -)Vergrößerung (Brennweite Teleskop/ Brennweite Okular),

-)Öffnungsverhältnis (Brennweite Teleskop/Objektivdurchmesser),

-) Auflösung : Mittelwert von $(115/\text{Objektivdurchmesser})$ und $B=(0,25164*528,88/\text{Objektivdurchmesser})$

```

class Teleskop
{
// Instance Variables
...
// Constructors
...
// Methods
...
}

```

```

import eprog.*;

class Teleskop
{
    double objektiv;
    double brennweiteOk;
    double brennweiteTe;
    Teleskop (double thebrennweiteOk, double thebrennweiteTe, double
theobjektiv)
    {
        objektiv=theobjektiv;
        brennweiteTe=thebrennweiteTe;
        brennweiteOk=thebrennweiteOk;
    }
    double vergroess()
    {
        return (brennweiteTe/brennweiteOk);
    }
    double oeffnung()
    {
        return (brennweiteTe/objektiv);
    }
    double aufloes()
    {
        return ((115/objektiv)+((0.25164*528.88)/objektiv))/2;
    }
}

```

b) Erstelle TeleskopTester welche Teleskop mit 3.0, 6.5 und 0.8 initialisiert und Vergrößerung, Öffnungsverhältnis und Auflösung ausgibt.

```

import eprog.*;

class TeleskopTester
{
    public static void main (String [] args)
    {

```

```
Teleskop tester = new Teleskop (3.0,6.5,0.8);  
EprogIO.println(tester.oeffnung());  
EprogIO.println(tester.vergroess());  
EprogIO.println(tester.aufloes());  
}  
}
```

Grün = Lösungen
Schwarz = Angabe