

Dr. Mathias J. Krause
M. Sc. Albert Mink
M. Sc. Zoltan Veszeka

1.2.2018

Einstieg in die Informatik und Algorithmische Mathematik

Klausur

Bearbeitungszeit: 90 min

Gesamtpunktzahl: 90 Punkte

Tragen Sie Ihren Vor- und Nachnamen, Ihren Studiengang und Ihre Matrikelnummer ein.

Vorname: _____

Nachname: _____

Studiengang: ☐ Bachelor Mathematik (alle) ☐ Sonst: _____

☐ Lehramt

Matrikelnummer: _____

Hinweise:

- Grundlage für diese Klausur ist die in der Vorlesung gelehrt Sprache Java.
- Es sind keine Hilfsmittel zugelassen.
- Jedes Lösungsblatt ist mit Namen und Matrikelnummer zu versehen.
- Schreiben Sie leserlich und mit einem dokumentenechten Stift.
(Lösungen in Bleistift, löschbarer Tinte usw. werden bei der Bewertung nicht berücksichtigt)
- Durchgestrichene Abschnitte werden bei der Bewertung ebenfalls nicht berücksichtigt.

Bewertung:

1	2	3	4	5	6	Σ	Note

Name, Matrikelnummer: _____

Aufgabe 1 *Ausdrücke*

(8 Punkte)

Geben Sie das Ergebnis und den Datentyp der nachfolgenden Ausdrücke an:

(a) $6/4$

Ergebnis: **Datentyp:**

(b) `((true && false) || (false || true))`

Ergebnis: **Datentyp:**

(c) `"Note " + "der Klausur"+"?"`

Ergebnis: **Datentyp:**

(d) $3/2.0$

Ergebnis: **Datentyp:**

(e) $3*1.5f$

Ergebnis: **Datentyp:**

(f) $4.0/2$

Ergebnis: **Datentyp:**

(g) `(char)('b'+2)`

Ergebnis: **Datentyp:**

(h) `((3>2) || (true || false) && (3<2))`

Ergebnis: **Datentyp:**

Name, Matrikelnummer:

Aufgabe 2 *Zahlensysteme*

(8 Punkte)

Geben Sie die entsprechende Lösung **mit Rechnung** an. Gesucht ist:

- (a) Die Dezimaldarstellung der Hexadezimalzahl $A9_{16}$.

- (b) Die Binärdarstellung der Dezimalzahl 153.

- (c) Die Hexadezimaldarstellung der Binärzahl 1110101001_2 .

- (d) Die Binärdarstellung der Hexadezimalzahl $5C9_{16}$.

Aufgabe 3 *Klassenhierarchie*

(2+4 Punkte)

Betrachten Sie die nachfolgenden Klassendefinitionen:

```
public class Gewässer {}  
  
public class Meere extends Gewässer {}  
  
public class Binnengewässer extends Gewässer {}  
  
public class Stillgewässer extends Binnengewässer {}  
  
public class Fließgewässer extends Binnengewässer {}  
  
public class Ozeane extends Meere {}
```

- (a) Erstellen Sie ein Klassendiagramm, welches die Beziehungen der Klassen untereinander veranschaulicht.

Fortsetzung auf der nächsten Seite

Name, Matrikelnummer:

(b) In der `main`-Methode eines Java-Programms stehen die Zeilen:

```
Gewässer      g;  
Ozeane        o;  
Fließgewässer fg;
```

Welche der folgenden Ausdrücke sind (unabhängig voneinander) gültig? Begründen Sie Ihre Entscheidung kurz.

(b1) `fg = new Meere();`

(b2) `g = new Meere();`

(b3) `o = new Gewässer();`

(b4) `fg = new Fließgewässer();`

Aufgabe 4 Effizienz

(3+5 Punkte)

(a) Gegeben ist folgender Programmausschnitt:

```
// Die Matrix a und der Vektor b sind gegeben
double x[n];
for (int i = n-1; i >= 0; i--) {
    double s = 0.0;
    for (int k = i; k < n; k++) {
        s += a[i][k] * x[k];
    }
    x[i] = 1/a[i][i] * (b[i]-s);
}
```

Zählen Sie die zur Berechnung nötige Anzahl an Rechenoperationen. Geben Sie den Rechenaufwand des Ausschnittes mit Hilfe eines Ausdrucks in der Landau-Notation an. Vereinfachen Sie diesen dann soweit wie möglich, z. B.. $\mathcal{O}(n^2 + 3n + 4) = \mathcal{O}(n^2)$.

(b) Vereinfachen Sie die folgenden Ausdrücke zur Komplexität durch geeignetes Runden, zu jeweils einem Ausdruck in der Landau-Notation, z. B.. $\mathcal{O}(n^3 + n) = \mathcal{O}(n^3)$:

1) $\mathcal{O}(4 * \ln(n) + 1/n + n^2/n)$

2) $\mathcal{O}((n^3 + 1) * n + n^2 + n * e^{10})$

3) $\mathcal{O}(1 + \ln(n) + 1/n^2)$

4) $\mathcal{O}(10n/(2n^2) + 10)$

5) $\mathcal{O}(10 * n^5 * n^3 + n * \ln(n) + 2^n/2)$

Aufgabe 5 Lottowahrscheinlichkeiten

(23 Punkte)

Der Binomialkoeffizient $\binom{n}{k}$ (lies n über k) ist für $n, k \in \mathbb{N}$ definiert als

$$\binom{n}{k} := \begin{cases} \frac{n!}{k!(n-k)!} = \frac{n \cdot (n-1) \cdots (n-k+1)}{1 \cdot 2 \cdots k} & \text{für } 0 \leq k \leq n, \\ 0 & \text{sonst.} \end{cases}$$

Schreiben Sie ein kompilierbares Java-Programm, das mit Hilfe der oben definierten Binomialkoeffizienten die Gewinnwahrscheinlichkeiten beim 6-er Lotto berechnet. Gehen Sie dazu wie folgt vor:

- Schreiben Sie eine Methode `biko` mit ganzzahligem Rückgabewert und zwei ganzzahligen formalen Parametern, die den Binomialkoeffizient $\binom{n}{k}$ für zwei natürliche Zahlen n und k mit $0 \leq k \leq n$ bestimmt. Setzen Sie dazu folgenden Algorithmus mit einer `for`-Schleife um:

$$c_0 := 1, \quad c_i := \frac{c_{i-1} \cdot (n - i + 1)}{i}, \quad \text{für } i = 1, \dots, k.$$

Dann gilt $c_k = \binom{n}{k}$.

- Schreiben Sie eine **rekursive** Methode `bikorek` mit ganzzahligem Rückgabewert und zwei ganzzahligen formalen Parametern, die den Binomialkoeffizienten rekursiv mit Hilfe der folgenden Formel bestimmt:

$$\binom{n}{k} = \begin{cases} \binom{n-1}{k-1} \cdot \frac{n}{k} & \text{für } 1 \leq k \leq n, \\ 1 & \text{für } k = 0, \\ 0 & \text{sonst.} \end{cases}$$

- Schreiben Sie nun das Hauptprogramm, in dem die Wahrscheinlichkeit

$$w(k) = \frac{\binom{6}{k} \cdot \binom{43}{6-k}}{\binom{49}{6}}$$

von $k = 0, 1, \dots, 6$ Richtigen im 6-er Lotto (6 aus 49) in einer `for`-Schleife mit den Methoden `biko` und `bikorek` berechnet werden. Speichern Sie dazu die Werte 6 und 49 vor der Schleife in entsprechende Variablen ab. Speichern Sie die jeweiligen Wahrscheinlichkeiten in je einem Feld vom Typ `double[]` ab und geben Sie diese mit begleitendem Text auf dem Bildschirm aus.

Aufgabe 6 *Tennisturnier*

(37 Punkte)

Schreiben Sie ein kompilier- und ausführbares Java-Programm, das ein einfaches Tennisturnier modelliert.

Verwenden Sie dazu die folgenden Klassen:

- Eine Klasse `Spieler`, die einen Spieler des Turniers simuliert. Diese können gegen andere Spieler antreten und dabei Preisgeld gewinnen, sowie die nächste Runde des Turniers erreichen.
- Eine Klasse `Turnier`, die den Ablauf des Turniers simuliert. Dabei wird das Teilnehmerfeld gefüllt und die ersten Spiele werden gespielt.

Gehen Sie wie folgt vor:

Schreiben Sie eine Klasse `Spieler`, welche die folgenden Elemente enthält:

- Eine ganzzahlige Variable `nummer` für die Identifikation des Spielers sowie zwei ganzzahlige Variablen `preisgeld` und `erreichte_runde` für das vom Spieler bis jetzt verdiente Preisgeld und den Fortschritt im Turnier. Außerdem eine Variable `turnier` vom Typ `Turnier`, die dem Spieler das gewählte Turnier zuweist. Dabei soll nur die Variable `nummer` als öffentliche Instanzvariable deklariert werden. Alle anderen Variablen sollen private Instanzvariablen sein.
- Einen öffentlichen Konstruktor mit zwei Parametern für die ganzzahlige Nummer des Spielers und das Turnier vom Typ `Turnier`, an dem der Spieler teilnimmt. Initialisieren Sie entsprechende Instanzvariablen mit den übergebenen Werten, lesen Sie das bis jetzt erreichte Preisgeld von der Konsole ein und belegen Sie die Variable `erreichte_runde` mit dem Wert 1.
- Eine öffentliche Instanzmethode `spiel` mit einem formalen Parameter für den Gegner (`SpielerB`) von `SpielerA` vom Typ `Spieler` und ganzzahligen Rückgabewert. Innerhalb der Methode sollen zunächst die Teilnehmer des Spiels mit begleitendem Text auf dem Bildschirm ausgegeben werden. Simulieren Sie nun den Ausgang des Spiels. Definieren Sie dazu eine Gleitkommazahl und belegen Sie diese mit einem zufälligen Wert zwischen 0 und 1. Falls diese Zahl kleiner als 0.5 ist, soll `SpielerA` das Spiel gewinnen, sonst `SpielerB`. Erhöhen Sie die `erreichte_runde` des Siegers um 1, addieren Sie zu dem Preisgeld des Siegers 1000 und geben Sie die Nummer des Siegers zurück.
Hinweis: Eine Zufallszahl zwischen 0 und 1 können Sie mit `Math.random()` erzeugen.
- Eine öffentliche Instanzmethode `infos` ohne Rückgabewert und ohne formalen Parameter. Geben Sie die Nummer des Spielers, die erreichte Runde und das gewonnene Preisgeld mit begleitendem Text auf dem Bildschirm aus.

Schreiben Sie eine weitere Klasse `Turnier`, welche folgende Elemente enthält:

- Eine ganzzahlige Variable `anzahlTeilnehmer` für die Anzahl der Teilnehmer des Turniers. Außerdem ein eindimensionales Feld `teilnehmer` vom Typ `Spieler`, in dem alle Teilnehmer des Turniers gespeichert werden. Beide Variablen sollen als öffentliche Instanzvariablen deklariert werden.

- Einen öffentlichen Konstruktor mit einem formalen Parameter für die ganzzahlige Anzahl der Teilnehmer des Turniers. Initialisieren Sie die entsprechende Instanzvariable mit dem übergebenen Wert und allokatieren Sie Speicher für das Feld. Iterieren Sie nun über die Anzahl der Spieler und legen Sie dabei die Spieler mit dem expliziten Aufruf des entsprechenden Konstruktors in dem Feld `teilnehmer` ab.
Hinweis: Die Nummer der Laufvariablen beim Füllen des Teilnehmerfeldes – beginnend mit Null – entspricht dabei der Nummer des Spielers.
- Das Hauptprogramm. Gehen Sie dabei wie folgt vor:
 - Erstellen Sie ein Turnier `turnier1` mit einem Teilnehmerfeld der Größe 8.
 - Simulieren Sie nun die erste Runde des Turniers. Dabei sollen mit einer Schleife vier Spiele simuliert werden. Der Spieler mit der Nummer `i` soll dabei gegen den Spieler mit der Nummer `7-i` spielen.
Geben Sie in jedem Schleifendurchlauf auch die Informationen des Siegers mit Hilfe der Methode `infos` mit begleitendem Text auf dem Bildschirm aus.

Name, Matrikelnummer:

Name, Matrikelnummer: _____

Lösung 1 *Ausdrücke*

Gesamt 8 Punkte, je 1 pro Aufgabenteil (0.5 Punkt für Ergebnis, 0.5 Punkt für Datentyp)

- (a) 1 (**int**)
- (b) true (**boolean**)
- (c) Note der Klausur? (**String**)
- (d) 1.5 (**double**)
- (e) 4.5 (**float**)
- (f) 2.0 (**double**)
- (g) d (**char**)
- (h) false (**boolean**)

Lösung 2 *Zahlensysteme*

Gesamt 8 Punkte. Je 2 pro Aufgabenteil (1 Punkt Ergebnis, 1 Punkt Rechnung)

- (a) $A9_{16} = 10 * 16 + 9 = 169$
- (b) $153 = 1 * 128 + 1 * 16 + 1 * 8 + 1 = 10011001_2$
- (c) $1110101001_2 = 3A9_{16}$
- (d) $5C9_{16} = 10111001001_2$

Name, Matrikelnummer: _____

Lösung 3 *Klassenhierarchie*

Gesamt 6 Punkte, Teil (a) 2 Punkte, Teil (b) jeweils 1.0 Punkte (Antwort: 0.5P, Begründung: 0.5P)
(b)

- (b1) Ungültig. Die Klasse `Fließgewässer` ist wie die Klasse `Meere` von der Basisklasse `Gewässer` abgeleitet. Jedoch ist `Fließgewässer` nicht Basisklasse der Klasse `Meere`. ("Meere sind keine Fließgewässer")
- (b2) Gültig. Der Variable `g` vom Typ `Gewässer` wird eine Instanz der Klasse `Meere` zugewiesen. ("Meere sind Gewässer")
- (b3) Ungültig. Die Klasse `Gewässer` ist Basisklasse der Klasse `Ozeane`. Daher ist eine Instanz der Klasse `Gewässer` keine Instanz der Klasse `Ozeane`. ("Gewässer sind im Allgemeinen keine Ozeane")
- (b4) Gültig. Der Variable `fg` vom Typ `Fließgewässer` wird eine Instanz der Klasse `Fließgewässer` zugewiesen. ("Fließgewässer sind Fließgewässer")

Lösung 4 Effizienz

Gesamt: 8 Punkte. Teil a) 2+1 Punkte (Begründung und Ergebnis), Teil b) 1 Punkt.

- (a) (1) *
- Die innere Schleife wird je $0, 1, 2, \dots, n-1$ -mal durchgelaufen. Dort wird je eine Addition und eine Multiplikation durchgeführt. Das sind $2 * (0 + 1 + 2 + \dots + (n-1))$ Operationen.
 - In der Äußeren Schleife wird noch bei jedem Schleifendurchlauf eine Division, eine Multiplikation und eine Subtraktion durchgeführt. Das entspricht $3n$ Operationen.
 - Insgesamt sind das (wegen der Gauß Formel) $2 * (0 + 1 + 2 + \dots + (n-1)) + 3n = 2 * ((n-1) * n/2) + 3n = n^2 - n + 3n = n^2 + 2n$ Operationen. Das entspricht einer Komplexität von $\mathcal{O}(n^2)$

- (b)
- $\mathcal{O}(4 * \ln(n) + 1/n + n^2/n) = \mathcal{O}(4 * \ln(n) + 1/n + n) = \mathcal{O}(n)$
 - $\mathcal{O}((n^3 + 1) * n + n^2 + n * e^{10}) = \mathcal{O}(n^4 + n + n^2 + n * e^{10}) = \mathcal{O}(n^4)$
 - $\mathcal{O}(1 + \ln(n) + 1/n^2) = \mathcal{O}(\ln(n))$
 - $\mathcal{O}(10n/(2n^2) + 10) = \mathcal{O}(5/n + 10) = \mathcal{O}(10) = \mathcal{O}(1)$
 - $\mathcal{O}(10 * n^5 * n^3 + n * \ln(n) + 2^n/2) = \mathcal{O}(10 * n^8 + n * \ln(n) + 2^n/2) = \mathcal{O}(2^n)$

Lösung 5 Lottowahrscheinlichkeiten

Lottowahrscheinlichkeiten.cpp

```
/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Klasse Lottowahrscheinlichkeiten                                %%
%% Berechnet mit Hilfe der Binomialkoeffizienten (iterativ und    %%
%% rekursiv die Gewinnwahrscheinlichkeit beim Lotto.             %%
%%                                                                %%
%% Vorlesung: 'Einfuehrung in die Informatik mit JAVA', WS 17     %%
%% Klausur, Aufgabe 4                                             %%
%%                                                                %%
%% Entwickelt am IANM 2, KIT                                       %%
%% Letzte Aenderung: Dez 2017, Z. Veszelka .                      %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/

import java.util.*;                                              // 1 | 23

public class Lottowahrscheinlichkeiten {                          // 1 | |

    public static int biko(int n, int k) {                        // 1 5 |
        int binkoeff = 1;                                        // 1 | |
        for(int i=1; i<=k; i++) {                                // 1 | |
            binkoeff = binkoeff * (n-i+1) / i;                  // 1 | |
        }                                                        // 1 | |
        return binkoeff;                                         // 1 | |
    }

    public static int bikorek(int n, int k) {                    // 1 6 |
        if (n < k || k < 0)                                     // 1 | |
            return 0;                                           // 0.5 | |
        if (k == 0)                                             // 1 | |
            return 1;                                           // 0.5 | |
        return bikorek (n-1, k-1) * n / k;                      // 2 | |
    }

    public static void main(String args[]) {                     // 1 10 |
        int n = 49, g = 6;                                     // 1,5 | |
        double [] wkeit1 = new double [g+1];                   // 1,5 | |
        double [] wkeit2 = new double [g+1];                   // | | |
        for (int k = 0; k <= g; k++) {                          // 1 | |
            wkeit1[k] = 1.0 * biko(g,k) * biko(n-g,g-k) / biko(n,g); // 1,5 | |
            System.out.println("Wkeit von "+k+" richtigen iterativ:"+wkeit1[k]); // 1 | |
            wkeit2[k] = 1.0 * bikorek(g,k) * bikorek(n-g,g-k) / bikorek(n,g); // 1.5 | |
            System.out.println("Wkeit von "+k+" richtigen rekursiv:"+wkeit2[k]); // 1 | |
        }
    }
}
```

Lösung 6 Tennisturnier

Tennisturnier.java

```
/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*** Klasse Tennisturnier ***
*** Simuliert zusammen mit der Klasse Spieler ein kleines ***
*** Tennisturnier ***
*** ***
*** Vorlesung: 'Einfuehrung in die Informatik mit JAVA', WS 17 ***
*** Klausur, Aufgabe 5 ***
*** ***
*** Entwickelt am IANM 2, KIT ***
*** Letzte Aenderung: Dez 2017, Z. Veszelka . ***
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/

import java.util.*; // 1 1 37

/*-----*/
/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*** Beginn Klasse. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
class Spieler { // 1 1 |

    /*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    *** Deklariere Instanz-Variablen. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
    public int nummer; // 1 3 |
    private int preisgeld, erreichte_runde; // 1 | |
    private Turnier turnier; // 1 | |

    /*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    *** Erstelle Konstruktor. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
    public Spieler(int n, Turnier turnier) { // 1 6 |
        nummer = n; // 1 | |
        this.turnier=turnier; // 1 | |
        Locale.setDefault(Locale.US); // 1 | |
        Scanner sc = new Scanner(System.in); // | | |
        System.out.println("Das von Spieler Nr." + nummer + " bis jetzt" // | | |
            + " gewonnene Preisgeld betraegt:"); // | | |
        preisgeld = sc.nextInt(); // 1 | |
        erreichte_runde = 1; // 1 | |
    }
}
```

Tennisturnier.java

```

/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*** Spieler spielt gegen einen Gegner %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
public int spiel(Spieler gegner) {
    System.out.println("Spieler Nr."+this.nummer + " spielt gegen "+
        "Spieler Nr."+ gegner.nummer);
    double a = Math.random();
    if (a<0.5){
        this.erreichte_runde++;
        this.preisgeld +=1000;
        return this.nummer;
    }
    else{
        gegner.erreichte_runde++;
        gegner.preisgeld += 1000;
        return gegner.nummer;
    }
}

/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*** Gebe Spieler-Informationen als Zeichenkette zurueck. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
public void infos () {
    System.out.println("Spieler Nr." + nummer + " hat die "
        +erreichte_runde+". Runde erreicht und hat " + preisgeld
        + " Preisgeld gewonnen\n");
}

}

/*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*** Beginn Klasse. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
class Turnier {
    /*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    *** Deklariere Instanz-Variablen. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
    public Spieler[] teilnehmer;
    public int anzahlTeilnehmer;
    /*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    *** Erstelle Konstruktor. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
    public Turnier (int anzahlTeilnehmer) {
        this.anzahlTeilnehmer = anzahlTeilnehmer;
        teilnehmer = new Spieler[anzahlTeilnehmer];
        for (int i=0; i<anzahlTeilnehmer; i++)
            teilnehmer[i] = new Spieler(i,this);
    }

    /*%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    *** Hauptprogramm %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%*/
    public static void main (String [] args) {
        Turnier turnier1 = new Turnier(8);
        for (int i=0; i<4; i++){
            int k=turnier1.teilnehmer[i].spiel(turnier1.teilnehmer[7-i]);
            System.out.println("Infos des Siegers der 1. Runde:");
            turnier1.teilnehmer[k].infos();
        }
    }
}

```