



**Abschlussprüfung
Java – Codeverständnis**
DATUM: ___ . ___ . ___ PROBEKLAUSUR

Matrikelnummer: _____

Raum: _____

Gruppe: _____

Punkte pro Aufgabe

Aufgabe	Maximale Punktzahl	Erreichte Punktzahl
1	3	
2	3	
3	4	
4	7	
5	3	
6	8	
7	11	
8	3	
9	8	

Gesamtpunktzahl: _____

Prozent: _____

Note: _____



Hinweise zur Klausur

- **Sie haben 60 Minuten Zeit.**
- **Es sind keine Hilfsmittel zugelassen.**

1) Was machen Directors und Clients beim Builder-Pattern? (3 Punkte)

2) Worin besteht das Konzept der javafx.beans.properties? (3 Punkte)

3) Nennen Sie die vier Vorteile, die sich ergeben, wenn Threads ordnungsgemäß verwendet werden. (4 Punkte)

4) Beschreiben Sie, was der Begriff „Repository“ im Git-Umfeld bedeutet und welche Funktion ein Repository hat. (7 Punkte)

5) Warum gibt es statische Variablen und Methoden in Java? Was unterscheidet eine Variable, die mit dem Attribut „static“ versehen wurde, von einer Variablen, die ohne dieses Attribut auskommen muss? (3 Punkte)

6) Erklären Sie mit Hilfe folgender Grafik, welche die Startseite der Java Docs im Browser zeigt, welche Bedeutung die drei Fenster haben, was in ihnen angezeigt wird und was passiert, wenn man bestimmte Elemente anklickt. (8 Punkte)

The screenshot shows the Java™ Platform, Standard Edition 8 API Specification. The left sidebar lists various Java packages and classes. The main content area displays the title "Java™ Platform, Standard Edition 8 API Specification" and a brief description: "This document is the API specification for the Java™ Platform, Standard Edition." Below this, there is a section titled "Profiles" with a list of three items: "compact1", "compact2", and "compact3". The right side of the main content area shows two entries under the heading "Profiles": "java.applet" and "java.awt". The "java.applet" entry has a detailed description: "Provides the classes necessary to create an applet and the classes an applet uses to communicate with its applet context." The "java.awt" entry also has a description: "Contains all of the classes for creating user interfaces and for painting graphics and images."

7) Gegeben ist ein unfertiges Programm, mit dem Objekte der Klasse „Dog“ mithilfe einer Standardimplementierung des BuilderPatterns erstellt werden können sollen. Sie haben folgende Aufgaben (**Gesamt: 11 Punkte**):

- Vervollständigen Sie in der Klasse „Dog“ den Konstruktor, in welchem der Name und das Alter des Hundes auf die bei der Implementierung eines BuilderPatterns übliche Weise gesetzt werden sollen. (2 Punkte)
- Vervollständigen Sie die Lücken in der Klasse „DogBuilder“. (3 Punkte)
- Vervollständigen Sie die Instanziierung des Objektes der Klasse „Dog“. Nutzen Sie hierfür die beim BuilderPattern übliche Instanziierungsform. (2 Punkte)
- Vervollständigen Sie danach die Klasse „Main“ und die Methode „wuff()“ der Klasse „Dog“ so, dass nachfolgende Ausgabe in der Konsole angezeigt wird. In der if-Bedingung soll dabei überprüft werden, ob die Variable „food“ des Objektes der Klasse „Dog“ false ist (4 Punkte):

Bello hat Hunger.
Der Fressnapf ist leer. -> Bello bellt.
Der Diener füllt den Fressnapf auf.

```
public class Main {

    public static void main(String[] args) {

        Dog dog = _____ .withName("Bello")
                        .withHasFood(false)._____;
        System.out.println(dog.getName()
                            + " hat Hunger.");
        if (_____ ) {
            _____ ("Der Fressnapf ist leer. ");
            _____ ;
            System.out.println("Der Diener füllt "
                                + "den Fressnapf auf.");
            dog.setHasFood(true);
        }
    }
}
```

Fortsetzung auf Seite 7

```
public class Dog {  
  
    private String name;  
    private boolean hasFood;  
  
    public Dog(DogBuilder builder) {  
        this.name = _____;  
        this.hasFood = _____;  
    }  
  
    public String getName() {  
        return this.name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }  
  
    public boolean getHasFood() {  
        return this.hasFood;  
    }  
  
    public void setHasFood(boolean hasFood) {  
        this.hasFood = hasFood;  
    }  
  
    public void wuff() {  
        System.out.println  
            ("-> " + _____ + " bellt.");  
    }  
}
```

Fortsetzung auf Seite 8

```
public class DogBuilder {  
  
    private String name;  
    private boolean hasFood;  
  
    public DogBuilder withName(String name) {  
        this.name = name;  
        _____;  
    }  
  
    public DogBuilder withHasFood(boolean hasFood) {  
        this.hasFood = hasFood;  
        _____;  
    }  
  
    public String getName() {  
        return this.name;  
    }  
  
    public boolean getHasFood() {  
        return this.hasFood;  
    }  
  
    public Dog build() {  
        _____;  
    }  
}
```

8) Wie lautet die Ausgabe des folgenden Programms? Falls mehr als fünf Zeilen ausgegeben werden, schreiben Sie in die sechste Zeile „usw“. (3 Punkte)

```
public class Main {  
  
    public static void main(String[] args) {  
  
        for (int i = 20; i > 0; i = i % 7 + 3) {  
            if (i == 7) {  
                break;  
            } else if (i % 3 == 1) {  
                continue;  
            }  
  
            System.out.println(--i);  
        }  
  
    }  
}
```

Ausgabe:

9) Welche vier Zeilen des folgenden Codes beinhalten Syntaxfehler? Nummerieren sie die Zeilen von 1 bis 4 und schreiben Sie den richtigen Code auf. Gehen Sie davon aus, dass es eine Klasse „TeamComperator“ gibt, welche „Comperator<Team>“ implementiert und einen Konstruktor ohne Übergabeparameter hat und eine Klasse „Team“ mit einem Konstruktor, dem ein String übergeben werden muss. In der Klasse „FootballLeagueView“ existiert eine Methode „configureStage“, welcher ein Objekt vom Typ „Stage“ übergeben werden muss. Gehen Sie außerdem davon aus, dass alle benötigten Imports vorhanden sind. **(8 Punkte)**

```

public class FootballLeagueView extends Application{

    private final ObservableList<Team>
        teams = FXCollections
            .observableArrayList(
                new Team("SV Sieghausen"),
                new Team("FC Fallenstetten"),
                new Team("TV Tormacher"),
                new Team("Eintracht Entenhausen"),
                new Team("SV Schwalbe"));

    public static void main(String[] args) {
        Launch(args);
    }

    @Override
    public void start(Stage stage) {
        TeamComparator comp = new TeamComparator();
        ObservableList<Team> sortedTeams =
            FXCollections
                .observableArrayList(this.teams);
        Collections.sort(sortedTeams, comp.reversed());
        Scene scene = new Scene(new Group());
        configureStage(stage);

        final Label label = new Label("Fußballliga");
        label.setFont("Arial", 24);
        final Label teamsLabel = new Label("Teams:");
        ComboBox<String> firstTeamBox = new ComboBox<>();
        for (Team team : teams) {
            firstTeamBox.add(team.getName());
        }
    }
}

```

Markierungsnummern der Zeilen mit Syntaxfehlern und richtiger Code:

1. _____
2. _____
3. _____
4. _____



**Abschlussprüfung
Java – Multiple Choice**
DATUM: ___ . ___ . ___ PROBEKLAUSUR

Matrikelnummer: _____

Raum: _____

Gruppe: _____

Punkte pro Aufgabe

Aufgabe	Maximale Punktzahl	Erreichte Punktzahl
1	2	
2	2	
3	2	
4	2	
5	2	
6	2	
7	2	
8	3	
9	2	
10	2	
11	4	

Gesamtpunktzahl: _____

Prozent: _____

Note: _____

Hinweise zur Klausur

- **Sie haben 60 Minuten Zeit.**
- **Es sind keine Hilfsmittel zugelassen.**

Aufgabe 1: a = true und b = false sind als boolean Variablen initialisiert. Welche der folgenden Zuweisungen haben den Wert true? **(2 Punkte)**

boolean x = !a && !a b;	
boolean x = b && a a;	
boolean x = a && b ^ a;	
boolean x = b ^ !a b;	

Aufgabe 2: Welche der folgenden Anweisungen werden vom Java-Compiler unter bestimmten Umständen akzeptiert? **(2 Punkte)**

if (i < 3) { i++; }	
return = i;	
if (alter >= 18) { System.out.println("Du bist Volljährig"); }	
return alter;	

Aufgabe 3: Gegeben ist folgende Codezeile:

```
float a = (int) 10;
```

Welche der folgenden Zuweisungen haben den Wert 0.5? **(2 Punkte)**

double b = a / 4 - 2;	
double b = a % 10 + 0.5;	
double b = a / 2.5 * 4;	
double b = a % 10 - 0.5;	

Aufgabe 4: Welche der folgenden Ausdrücke liefern true zurück? **(2 Punkte)**

“car” == “car”	
new String (“car”) == “car”	
new String (“car”) == new String (“car”)	
“carx” == “carx”	

Aufgabe 5: Welche 2 Aussagen treffen in Bezug auf den String-Pool zu? **(2 Punkte)**

Für jeden String, der mit Anführungszeichen definiert wird, legt Java ein Objekt an, merkt es sich aber nicht im so genannten String-Pool.	
Verwenden wir einen String mit demselben Inhalt noch einmal, dann prüft Java, ob es diesen String bereits im String-Pool gibt und verwendet ihn ggf. wieder.	
Wir können mit Sicherheit immer davon ausgehen, dass Java den String-Pool in allen Situationen verwendet.	
Sollten wir einen String mittels new erzeugen, dann missachtet Java den String-Pool.	

Aufgabe 6: Welche Evaluierungen gibt es bei Bindings? **(2 Punkte)**

Ambitious Evaluation	
Eager Evaluation	
Rotten Evaluation	
Lazy Evaluation	

Aufgabe 7: Welche Zustände werden bei Threads unterschieden? **(2 Punkte)**

OLD	
NEW	
PAUSING	
WAITING	

Aufgabe 8: Welche Methoden stellt die Klasse Thread bereit? **(3 Punkte)**

sleep()	
sleep(long millis)	
wasInterrupted()	
isInterrupted()	
interrupt()	
awake()	

Aufgabe 9: Welche Aussagen treffen auf den folgenden Code zu? **(2 Punkte)**

```
public class SimpleTest {  
  
    public static void main(String[] args) {  
        int[] table = { 1, 2, 3, 4, 5 };  
        table[1] = (table[2 * 1] == 2 - args.length) ? table[3] : 99;  
        System.out.println(table[1]);  
    }  
}
```

Es wird folgendes auf der Konsole ausgegeben: 3	
Es wird folgendes auf der Konsole ausgegeben: 99	
Der Code ist kompilierbar	
Der Code ist nicht kompilierbar	

Aufgabe 10: Gegeben sind einige Codeschnipsel. Welche der folgenden Codeschnipsel sind fehlerhaft und verhindern eine Kompilierung? Exceptions gelten hier nicht als Fehler. Gehen Sie davon aus, dass alle benötigten Imports vorhanden sind. **(2 Punkte)**

```
public static void main(String[] args) {  
    int x = 0;  
    int y = 1354 / x;  
}
```

```
public static void main(String[] args) {  
    int b = 1;  
    int c = 1;  
    do {  
        System.out.println("x");  
    } while (b = c);  
}
```

```
public static void main(String[] args) {  
    char a = 'a';  
    char c = 'b';  
    c = a;  
}
```

```
public static void main(String[] args) {  
    double a = (float)35.14;  
    double b = a;  
    double c = (double)a - (float)b;  
    float d = c;  
}
```

Aufgabe 11: Das folgende Programm enthält mehrere Syntax- und Konventionsfehler. Finden Sie zwei Syntax- und zwei Konventionsfehler und geben Sie jeweils die Zeile und eine kurze Erläuterung an. Logikfehler (wie z.B. Endlosschleifen) gelten hier NICHT als Fehler. Falls Sie mehr als die geforderten Fehler angeben, werden nur die ersten zwei Syntax- und die ersten zwei Konventionsfehler gewertet. Fehler, die aus anderen Fehlern resultieren, werden nur als ein Fehler gewertet. Für jede richtige Zeile und jede richtige Erläuterung gibt es jeweils 0,5 Punkte. **(4 Punkte)**

```
1
2 public class MistakesN {
3
4     public static void main(String[] args) {
5         int a = 5;
6         for (int i = 0; i < args.length) {
7             a %= 2;
8             if (a > 0) {
9                 System.out.println(i . a);
10            }
11            i++;
12        }
13    }
14
15 }
```

Syntaxfehler:

1. Zeile: Grund:

2. Zeile: Grund:

Konventionsfehler:

3. Zeile: Grund:

4. Zeile: Grund: