

KLAUSUR**PROGRAMMIERUNG 1****19. MÄRZ 2018**

Bedingungen der Klausur:

1. Wählen Sie einen neuen Workspace! Es darf nur das in dieser Klausur bearbeitete Projekt in Ihrem Workspace enthalten sein! Ein weiteres Projekt in Ihrem Workspace führt zu einem automatischen Ausschluss aus der Bewertung!
2. Die Nutzung von Handys, Abschreiben oder Kommunikation mit anderen während der Klausur ist nicht zulässig! Handys müssen ausgeschaltet und in der Tasche sein.
3. Ein handgeschriebenes A4-Blatt ist zugelassen.
4. Eclipse und ein Browser sind die einzigen Anwendungen, die auf Ihrem Rechner geöffnet sein dürfen. Im Browser dürfen Sie die Java-API-Dokumentation (<http://docs.oracle.com/javase/8/docs/api/>) und die Moodle-Seite für die Klausur-Aufgabe geöffnet sein. Jede andere geöffnete Seite führt zu einem automatischen Ausschluss aus der Bewertung!
5. Am Ende der Prüfung: Öffnen Sie den Dateexplorer, wechseln Sie in Ihren Workspace, zippen Sie Ihr gesamtes Projekt und laden Sie die zip-Datei in Moodle hoch (bei Aufgabe Klausur1)! Sie können Ihre java-Klassen auch einzeln hochladen.
6. Es sind insgesamt 65 Punkte zu erzielen (Teil 1: 11 Punkte, Teil 2: 10 Punkte, Teil 3: 18 Punkte, Teil 4: 23 Punkte + ausführbares, fehlerfreies Programm: 3 Punkte). Es wird von einer Gesamtpunktzahl von 62 Punkten ausgegangen. Zum Bestehen der Klausur sind 31 Punkte notwendig.

Notenspiegel:

| | | | | | | | | | | | |
|--------|-----|------|------|------|------|------|------|------|------|------|-----|
| Punkte | >58 | 55.0 | 52.0 | 49.0 | 46.0 | 43.0 | 40.0 | 37.0 | 34.0 | 31.0 | <31 |
| | | – | – | – | – | – | – | – | – | – | |
| | | 57.5 | 54.5 | 51.5 | 48.5 | 45.5 | 42.5 | 39.5 | 36.5 | 33.5 | |
| Note | 1,0 | 1,3 | 1,7 | 2,0 | 2,3 | 2,7 | 3,0 | 3,3 | 3,7 | 4,0 | 5,0 |

Vorbereitung:

1. Wechseln Sie in einen **leeren** Workspace!
2. Legen Sie ein neues Java-Projekt mit dem Namen **KlausurIhrName** an!
3. Schreiben Sie Ihre Klassen im package **klausur**!

Teil 1 (Klasse Pizza)**11 Punkte**

| | |
|--|--------|
| Schreiben Sie eine Klasse Pizza . | 1 Pkt. |
| Diese Klasse hat zwei Objektvariablen: name vom Typ String und preis vom Typ float . Beide Variablen sind nur innerhalb der Klasse sichtbar! | 2 Pkt. |
| Schreiben Sie eine statische Methode pizzaErzeugen(String name, float preis) . Diese Methode gibt ein Objekt vom Typ Pizza zurück. Den Objektvariablen des zurückgegebenen Pizza-Objektes werden die Werte der Parameter zugewiesen. | 2 Pkt. |
| Schreiben Sie Getter für die beiden Objektvariablen, also getName() und getPreis() . | 2 Pkt. |
| <p>Überschreiben Sie die toString() -Methode, so dass diese einen String der Form</p> <p>Die Pizza Salami kostet 6.9 Euro.</p> <p>zurückgibt (Beispielwerte für name und preis).</p> <p>Schreiben Sie außerdem eine print() -Methode, die den von toString() erzeugten String auf die Konsole ausgibt.</p> | 2 Pkt. |
| <p>Schreiben Sie eine Klasse KlausurMain mit main() -Methode.</p> <p>Erstellen Sie in der main() -Methode mithilfe der pizzaErzeugen() -Methode folgende 5 Pizza-Objekte (siehe gewünschte Ausgabe). Rufen Sie jeweils die print() -Methode auf. Es soll folgende Ausgabe entstehen:</p> <p>Die Pizza Salami kostet 6.9 Euro. Die Pizza Margheritha kostet 5.9 Euro. Die Pizza Tonno kostet 6.9 Euro. Die Pizza Hawaii kostet 6.9 Euro. Die Pizza Calzone kostet 7.9 Euro.</p> | 2 Pkt. |

Teil 2 (Klasse Pizzaverkauf)

10 Punkte

| | |
|--|--------|
| <p>Schreiben Sie eine Klasse Pizzaverkauf.</p> <p>Diese Klasse hat zwei Objektvariablen pizza vom Typ Pizza und anzVerkaeufe vom Typ int. Beide Variablen sind nur in der Klasse sichtbar.</p> | 1 Pkt. |
| <p>Schreiben Sie einen parametrisierten Konstruktor Pizzaverkauf(Pizza pizza). Mit dem Parameterwert pizza wird die Objektvariable pizza initialisiert. Der Wert der Objektvariablen anzVerkaeufe wird auf 0 gesetzt.</p> | 1 Pkt. |
| <p>Schreiben Sie eine Methode verkaufen(). Darin wird der Wert der Objektvariablen anzVerkaeufe um 1 erhöht.</p> | 1 Pkt. |
| <p>Schreiben Sie für die Objektvariablen Getter, also getAnzVerkaeufe() und getPizza().</p> | 1 Pkt. |
| <p>Schreiben Sie eine Methode umsatz(). Diese Methode gibt ein double zurück. Der Wert berechnet sich aus der Anzahl der Verkäufe der Pizza (anzVerkaeufe) mal dem Preis der Pizza.</p> | 2 Pkt. |
| <p><u>Überschreiben</u> Sie eine Methode toString(), so dass folgender String zurückgegeben wird (Beispielwerte).</p> <p>Pizza Salami wurde 0 mal zum Preis von 6.9 Euro verkauft.</p> <p>Schreiben Sie eine Methode print(), die den Wert von toString() auf die Konsole ausgibt.</p> | 2 Pkt. |
| <p>Erzeugen Sie in der main()-Methode der Klasse KlausurMain ein Objekt der Klasse Pizzaverkauf. Verwenden Sie dazu eines der fünf in Teil 1 erstellten Pizza-Objekte.</p> <ul style="list-style-type: none"> • Rufen Sie die Methode print() auf. • Rufen Sie die Methode verkaufen() auf. • Rufen Sie erneut die Methode print() auf. Es entsteht folgende Ausgabe: <p>Pizza Salami wurde 0 mal zum Preis von 6.9 Euro verkauft. Pizza Salami wurde 1 mal zum Preis von 6.9 Euro verkauft.</p> | 2 Pkt. |

Teil 3 (Klasse Speisekarte)

18 Punkte

| | |
|--|--------|
| <p>Schreiben Sie eine Klasse Speisekarte.</p> <p>Diese Klasse hat eine Objektvariable angebot vom Typ Pizza []. Diese Variable ist innerhalb des Paketes sichtbar.</p> | 1 Pkt. |
| <p>Schreiben Sie einen parameterlosen Konstruktor. In diesem Konstruktor wird für angebot ein Pizza-Array der Länge 0 erzeugt.</p> | 1 Pkt. |
| <p>Schreiben Sie eine Methode pizzaHinzufuegen (Pizza pizza).</p> <p>Diese Methode fügt die als Parameter übergebene pizza dem angebot-Array hinzu. Beachten Sie:</p> <ul style="list-style-type: none"> • Um dem angebot-Array eine neue Pizza hinzuzufügen, muss die Länge des Arrays um 1 erhöht werden. • Kopieren Sie sich dazu das alte angebot-Array. • Erzeugen Sie dann ein neues angebot-Array, das um 1 länger ist als das alte. • Kopieren Sie das bisherige Angebot zurück in das angebot-Array. • Fügen Sie die neue Pizza (Parameter pizza) als letztes Element im neuen angebot-Array hinzu. | 4 Pkt. |
| <p>Schreiben Sie eine Methode pizzaIstImAngebot (Pizza pizza). Diese Methode gibt ein true zurück, wenn die als Parameter übergebene pizza im angebot-Array enthalten ist. Prüfen Sie die Gleichheit der Namen der Pizza. Wenn die Pizza nicht im angebot-Array enthalten ist, gibt die Methode false zurück.</p> | 3 Pkt. |
| <p>Schreiben Sie eine Methode pizzaLoeschen (Pizza pizza).</p> <p>Diese Methode löscht die als Parameter übergebene pizza aus dem angebot-Array (wenn Sie darin enthalten ist). Beachten Sie:</p> <ul style="list-style-type: none"> • Nach dem Löschen der Pizza aus dem angebot-Array soll das angebot-Array wieder um 1 kleiner sein als vorher (falls die zu löschende Pizza überhaupt im angebot-Array enthalten war). • Kopieren Sie also das alte angebot-Array außer die zu löschende Pizza. • Ihre Kopie ist dann das neue angebot-Array. | 4 Pkt. |
| <p>Schreiben Sie eine Methode getPizzaAtIndex (int index). Diese Methode gibt die Pizza zurück, die im angebot-Array beim Index index eingetragen ist.</p> <p>Prüfen Sie, ob der übergebene Parameter ein gültiger Index aus dem angebot-Array ist. Wenn nicht, geben Sie null zurück.</p> | 2 Pkt. |
| <p>Schreiben Sie eine Methode print (). Diese Methode gibt die Speisekarte in tabellarischer Form wie folgt aus (Beispielwerte):</p> | |

| | |
|---|--------|
| <pre>===== Speisekarte ===== Salami 6,90 Euro Margheritha 5,90 Euro Tonno 6,90 Euro Hawaii 6,90 Euro Calzone 7,90 Euro</pre> | |
| <p>a) Erzeugen Sie in der <code>main()</code>-Methode der Klasse <code>KlausurMain</code> ein Objekt der Klasse <code>Speisekarte</code>. Fügen Sie unter Verwendung der Methode <code>pizzaHinzufuegen()</code> die fünf in Teil 1 erstellten Pizza-Objekte der Speisekarte hinzu. Rufen Sie die Methode <code>print()</code> auf. Es entsteht folgende Ausgabe:</p> <pre>===== Speisekarte ===== Salami 6,90 Euro Margheritha 5,90 Euro Tonno 6,90 Euro Hawaii 6,90 Euro Calzone 7,90 Euro</pre> <p>b) Löschen Sie nun mit Hilfe der <code>pizzaLoeschen()</code>-Methode die „Tonno“-Pizza (das dritte Pizza-Objekt aus Teil 1). Rufen Sie erneut die <code>print()</code>-Methode auf. Es entsteht folgende Ausgabe:</p> <pre>===== Speisekarte ===== Salami 6,90 Euro Margheritha 5,90 Euro Hawaii 6,90 Euro Calzone 7,90 Euro</pre> | 3 Pkt. |

Teil 4 (Klasse Pizzeria)

23 Punkte

| | |
|---|---------------|
| <p>Schreiben Sie eine Klasse Pizzeria. Objektvariable ist verkaeufe vom Typ Pizzaverkauf [].</p> | <p>1 Pkt.</p> |
| <p>Schreiben Sie einen parametrisierten Konstruktor Pizzeria (Speisekarte karte).</p> <ul style="list-style-type: none"> • Das verkaeufe-Array wird erzeugt und hat die gleiche Länge wie das angebot-Array der Speisekarte karte. • Jedes Element des verkaeufe-Arrays zeigt auf ein Pizzaverkauf-Objekt. Erzeugen Sie alle Pizzaverkauf-Objekte. Übergeben Sie dem Pizzaverkauf-Konstruktor dazu die jeweiligen Pizza-Objekte aus der Speisekarte. | <p>3 Pkt.</p> |
| <p>Schreiben Sie eine Methode bestellen (). Diese Methode gibt ein int zurück. In dieser Methode soll zufällig ein Index aus dem verkaeufe-Array erzeugt werden.</p> <p>Nutzen Sie dazu die Klasse Random aus dem java.util-Paket. Verwenden Sie die Objektmethode nextInt (int bound) der Klasse Random. Wenden Sie nextInt () so an, dass auch tatsächlich ein gültiger Index des verkaeufe-Arrays erzeugt wird.</p> | <p>2 Pkt.</p> |
| <p>Schreiben Sie eine Objektmethode verkaufen (int index). Durch den Aufruf der Methode wird die Pizza verkauft, die im verkaeufe-Array am Index index steht. Nutzen Sie für den Verkauf die verkaufen ()-Methode der Klasse Pizzaverkauf.</p> <p>Beachten Sie: überprüfen Sie, ob der als Parameter übergebene Wert für index tatsächlich einem Index im verkaeufe-Array entspricht.</p> | <p>2 Pkt.</p> |
| <p>Schreiben Sie eine Objektmethode tagesVerkauf (int anzVerkaeufe). In dieser Methode wird anzVerkaeufe oft eine Pizza verkauft (verkaufen (int index)). Welche Pizza verkauft wird (also welcher index), wird durch die Methode bestellen () jeweils zufällig ermittelt.</p> | <p>2 Pkt.</p> |
| <p>Schreiben Sie eine Objektmethode print (). Diese Methode erzeugt folgende Ausgabe (Beispielwerte):</p> <pre> Salami : ***** Margheritha : ***** Hawaii : ***** Calzone : ***** </pre> <p>Das heißt, es wird am Anfang der Zeile der Name der Pizza aus dem verkaeufe-Array ausgegeben und danach für die Anzahl der Verkäufe jeweils ein *.</p> | <p>4 Pkt.</p> |
| <p>Schreiben Sie eine Objektmethode meistverkauftePizza (). Diese Methode gibt die Pizza aus dem verkaeufe-Array zurück, die am meisten verkauft wurde.</p> | <p>4 Pkt.</p> |

| | |
|--|---------------|
| <p>Schreiben Sie eine Objektmethode gesamtEinnahmen () . Diese Methode gibt die Summe aller Einnahmen als double zurück. Die Einnahmen ergeben sich aus der Summe der Umsätze aller Pizzen (Methode umsatz () von Pizzaverkauf) aus dem verkaeufe-Array.</p> | <p>2 Pkt.</p> |
| <p>Erzeugen Sie in der main () -Methode der Klasse KlausurMain ein Objekt der Klasse Pizzeria . Übergeben Sie dem Pizzeria-Objekt die Speisekarte aus Teil 3. Rufen Sie die Methode tagesVerkauf () mit 150 Verkäufen auf. Rufen Sie die print () -Methode für das Pizzeria-Objekt auf. Es entsteht folgende Ausgabe (Beispielwerte):</p> <pre> Salami : ***** Margheritha : ***** Hawaii : ***** Calzone : ***** </pre> <p>Rufen Sie die Methoden meistverkauftePizza () und gesamtEinnahmen () so auf, dass folgende Ausgabe entsteht (Beispielwerte):</p> <pre> Meistverkaufte Pizza : Die Pizza Salami kostet 6.9 Euro. Die Gesamteinnahmen betragen 1023,00 Euro </pre> | <p>3 Pkt.</p> |

Zusammenfassung:

Mögliche Ausgaben (Werte können variieren)

----- Teil 1 -----

Die Pizza Salami kostet 6.9 Euro.
Die Pizza Margheritha kostet 5.9 Euro.
Die Pizza Tonno kostet 6.9 Euro.
Die Pizza Hawaii kostet 6.9 Euro.
Die Pizza Calzone kostet 7.9 Euro.

----- Teil 2 -----

Pizza Salami wurde 0 mal zum Preis von 6.9 Euro verkauft.
Pizza Salami wurde 1 mal zum Preis von 6.9 Euro verkauft.

----- Teil 3 -----

===== Speisekarte =====

| | |
|-------------|-----------|
| Salami | 6,90 Euro |
| Margheritha | 5,90 Euro |
| Tonno | 6,90 Euro |
| Hawaii | 6,90 Euro |
| Calzone | 7,90 Euro |

===== Speisekarte =====

| | |
|-------------|-----------|
| Salami | 6,90 Euro |
| Margheritha | 5,90 Euro |
| Hawaii | 6,90 Euro |
| Calzone | 7,90 Euro |

----- Teil 4 -----

| | |
|-------------|---------|
| Salami | : ***** |
| Margheritha | : ***** |
| Hawaii | : ***** |
| Calzone | : ***** |

Meistverkaufte Pizza : Die Pizza Margheritha kostet 5.9 Euro.
Die Gesamteinnahmen betragen 1018,00 Euro

Klassendiagramm:

| Pizza | |
|--|----------|
| - name | : String |
| - preis | : float |
| <hr/> | |
| + <u>pizzaErzeugen</u> (String, float) | : Pizza |
| + <u>getPreis</u> () | : float |
| + <u>getName</u> () | : String |
| + <u>toString</u> () | : String |
| + <u>print</u> () | |

| Speisekarte | |
|-----------------------------------|-----------|
| angebot | : Pizza[] |
| <hr/> | |
| + <u>Speisekarte</u> () | |
| + <u>pizzaHinzufuegen</u> (Pizza) | |
| + <u>pizzaIstAngebot</u> (Pizza) | : boolean |
| + <u>pizzaLoeschen</u> (Pizza) | |
| + <u>getPizzaAtIndex</u> (int) | : Pizza |
| + <u>print</u> () | |

| Pizzaverkauf | |
|-------------------------------|----------|
| - anzVerkaeufe | : int |
| - Pizza | : Pizza |
| <hr/> | |
| + <u>Pizzaverkauf</u> (Pizza) | |
| + <u>verkaufen</u> () | |
| + <u>getAnzVerkaeufe</u> () | : int |
| + <u>getPizza</u> () | : Pizza |
| + <u>Umsatz</u> () | : double |
| + <u>toString</u> () | : String |
| + <u>print</u> () | |

| Pizzeria | |
|---------------------------------|------------------|
| verkaeufe | : Pizzaverkauf[] |
| <hr/> | |
| + <u>Pizzeria</u> (Speisekarte) | |
| + <u>bestellen</u> () | : int |
| + <u>verkaufen</u> (int) | |
| + <u>tagesVerkauf</u> (int) | |
| + <u>print</u> () | |
| + <u>meistverkauftePizza</u> () | : Pizza |
| + <u>gesamtEinnahmen</u> () | : double |