Klausurprüfung (Fachtheorie)  
aus Programmieren und Software Engineering

# Coruscant 7 aus 42

Für Coruscant soll ein Lotteriespiel ähnlich dem Spiel “6 aus 45” eingeführt werden. Nachdem Sie mit dem österreichischen Lotteriespiel „6 aus 45“ und der Entwicklung von Webanwendungen vertraut sind werden Sie beauftragt die benötigte Software zu entwickeln.

## *Rahmenbedingungen*

* Die Applikation ist als Webanwendung zu implementieren.
* Die Benutzerschnittstelle ist als HTML Anwendung für aktuelle Browser zu implementieren.
* Für die gestalterische Aufbereitung der Anwendung ist ein Team von Grafikern beauftragt – Ihre Implementierung soll sich vorerst ausschließlich darum kümmern, dass die Gestaltung der Anwendung einfach möglich ist (Stichwort: Trennung von „Was wird dargestellt“ von „Wie wird es dargestellt“).
* Die Implementierung der Benutzerschnittstelle soll alle beschriebenen Geschäftsfällen ermöglichen (soll sich aber auf die Funktionalität beschränken).
* Alle relevanten Daten der unten angeführten Geschäftsfälle sind in einer relationalen Datenbank zu speichern.
* Achten Sie auf eine hohe Qualität bei der Implementierung!

## *Abzubildende Geschäftsfälle*

1. Im ersten Schritt wird nur eine Spielrunde implementiert – alle abgegebenen Tipps sind Teil derselben Spielrunde und nehmen an derselben und einzigen Ziehung teil. Die Verwaltung mehrerer Spielrunden ist NICHT Teil dieser ersten Implementierung.
2. Auch eine Verwaltung und Abrechnung der Annahmenstellen ist derzeit NICHT Teil der Implementierung
3. Ein Tipp besteht aus sieben Zahlen zwischen 1 und 42. Jede Zahl kann in einem Tipp nur einmal vorkommen. Sollten mehrere Tipps gemeinsam abgegeben werden so können sich in unterschiedlichen Tipps Zahlen wiederholen.
4. Der Kunde geht in eine Annahmestelle um mehrere Tipps abzugeben. Je Tipp kann der Kunde 7 Zahlen bekanntgeben. Als Ergebnis bekommt der Kunde eine Referenznummer für seine Tipps.
5. Der Kunde kann auch einen (oder mehrere) zufällig erzeugte Tipps abgeben – sogenannte Quicktipps. Die (sieben) gespielten Zahlen werden dabei mittels eines Zufallsgenerators erzeugt.
6. Dem Kunden sind für die abgegebenen Tipps eine Referenz und ein Preis zu ermitteln – je Tipp sind 3,7 Novian Ruby zu berechnen
7. Der Kunde kann – durch Eingabe der erhaltenen Referenz - seine Tipps einsehen.
8. Es muss möglich sein die sieben Gewinnzahlen der aktuellen Spielrunde einzugeben
9. Es muss möglich alle Tipps zu ermitteln die vier, fünf, sechs oder sieben richtige Tipps abgegeben haben
10. Es gibt eine fixe Zuteilung zwischen Gewinnrang und Einnahmen
    1. Gewinnrang vier richtige Zahlen: 5% der Einnahmen
    2. Gewinnrang fünf richtige Zahlen: 6% der Einnahmen
    3. Gewinnrang sechs richtige Zahlen: 12% der Einnahmen
    4. Gewinnrang sieben richtige Zahlen: 23% der Einnahmen
11. Der Gewinn je Gewinnrang wird dann noch durch die Anzahl der Tipps aus diesem Gewinnrang dividiert um den tatsächlich auszuzahlenden Betrag zu ermitteln
12. Ein Kunde kommt nach der Ziehung zur Annahmestelle und kann mit seiner Referenznummer die folgenden Informationen abrufen:
    1. Welcher seiner Tipps hat gewonnen
    2. In welchem Gewinnrang befinden sich seine Tipps
    3. Welche Gewinnsumme ist für diese Tipps auszuzahlen

## *Realisierungeschritte*

1. Erstellen Sie ein Klassenmodell in UML, welches den oben beschriebenen Geschäftsablauf abdeckt. Achten Sie dabei auf die zu verwendenden Datentypen und die Beziehungen zwischen den Klassen
2. Realisieren Sie die Anwendung in einer adäquaten Architektur (z.B. bei einer Mehrschichtarchitektur mit Domain, Persistence, Service und Presentation Schicht)
3. Realisieren Sie das Speichern und Lesen der Daten mittels transparentem OR Mapping.
4. Testen Sie die Bestandteile ihrer Anwendung um die Korrektheit der Implementierung zu beweisen (Achtung! Getesteter Code zählt mehr als ungetesteter)
5. Testen Sie die implementierten Geschäftsfälle
6. Erstellen Sie die Benutzeroberfläche für die beschriebenen Geschäftsfälle als HTML Seiten – folgende HTML Seiten sollten damit zumindest vorhanden sein:
   1. Eingabe von Tipps durch den Kunden
   2. Ergebnisseite der Tippeingaben mit Referenznummer und zu zahlendem Einsatz
   3. Eingabe der sieben Gewinnzahlen
   4. Statistik mit Gesamtsumme der Einnahmen, Anzahl der abgegebenen Tipps, Anzahl der Tipps je Gewinnrang, ermittelter Gewinn für einen Tipp aus jedem Gewinnrang (innerhalb eines Gewinnranges bekommen ja alle Tipps den selben Gewinn)
   5. Ergebnisabfrage des Kunden mittels Referenz – anzuzeigen sind: Gewinnzahlen, abgegebener Tipp, Gewinnsumme je Tipp

## *Realisierung*

Die Aufgabenstellung kann mittels:

* Java Enterprise Edition (Spring)
* ASP.NET (C#) oder
* JavaScript (NodeJS)

realisiert werden.

## *Bewertung*

Die Bewertung berücksichtigt sowohl welche Geschäftsfälle implementiert wurden als auch die Qualität der Implementierung.

Die erreichbaren Punkte je Geschäftsfall sind:

Geschäftsfall 1: nicht zu implementieren

Geschäftsfall 2: nicht zu implementieren

Geschäftsfall 3: 14

Geschäftsfall 4: 19

Geschäftsfall 5: 19

Geschäftsfall 6: 19

Geschäftsfall 7: 14

Geschäftsfall 8: 14

Geschäftsfall 9: 27

Geschäftsfall 10: 27

Geschäftsfall 11: 14

Geschäftsfall 12: 27

Die Punkte werden je nach Qualität vergeben – unterschieden wird je Geschäftsfall wie folgt:

L1 Die in der Geschäftsfallbeschreibung geforderte Funktionalität ist vorhanden  
50% der zu erreichenden Punkte

L2 Die geforderte Funktionalität ist durch Tests sicher gestellt (Testüberdeckung, Gut-Fälle sind abgedeckt, mögliche Fehlerfälle sind abgedeckt)  
~20% der zu erreichenden Punkte

L3 Die Prinzipien der objektorientierten Implementierung wurden umgesetzt (z.B. single responsibility, don’t repeat yourself, naming of classes and methods, encapsulation, coupling)  
~20% der zu erreichenden Punkte

L4 Stil und Lesbarkeit des Codes (z.B. einheitliche Formatierung, Namengebung, kein überflüssiger Code)  
~5% der zu erreichenden Punkte

Folgende Matrix bringt beide Aspekte zusammen:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Geschäftsfall** | **L1** | **L2** | **L3** | **L4** | **Summe** |
| Geschäftsfall 1 | Nicht zu implementieren | | | | |
| Geschäftsfall 2 | Nicht zu implementieren | | | | |
| Geschäftsfall 3 | 7 | 3 | 3 | 1 | 14 |
| Geschäftsfall 4 | 10 | 4 | 4 | 1 | 19 |
| Geschäftsfall 5 | 10 | 4 | 4 | 1 | 19 |
| Geschäftsfall 6 | 10 | 4 | 4 | 1 | 19 |
| Geschäftsfall 7 | 7 | 3 | 3 | 1 | 14 |
| Geschäftsfall 8 | 7 | 3 | 3 | 1 | 14 |
| Geschäftsfall 9 | 14 | 6 | 6 | 1 | 27 |
| Geschäftsfall 10 | 14 | 6 | 6 | 1 | 27 |
| Geschäftsfall 11 | 7 | 3 | 3 | 1 | 14 |
| Geschäftsfall 12 | 14 | 6 | 6 | 1 | 27 |

Die verbleibenden Punkte werden anhand des Gesamteindrucks der Abgabe vergeben. Kriterien hierfür sind z.B.

* die gewählte Architektur
* die Organisation (Struktur) des Projektes
* die Klarheit der Lösung im Gesamten
* die resultierenden HTML Seiten