

Quiz (35 Punkte)

Mein Code erzeugte beim Einreichen 2 Fehlermeldungen, die ich nicht wegbekam, funktioniert aber grundsätzlich so wie er soll. Wenn ihr die Fehler findet, gerne im Kommentar was dazu sagen.

In dieser Aufgabe sollen Sie ein kleines Ratequiz implementieren. Dazu werden die in Abbildung 1 dargestellten Klassen benötigt.

Bitte beachten Sie: Die automatische Korrektur unterliegt technischen Beschränkungen, so dass die Verarbeitung von mehreren als `public` deklarierten Klassen nicht möglich ist. Achten Sie bitte bei Ihren Einreichungen im Online-Übungssystem darauf, dass Ihre Klassen nicht als `public` deklariert sind.

Die einzelnen Schnittstellen und Klassen sind nachfolgend beschrieben. Sie dürfen die Klassen und Schnittstellen um beliebige Methoden und Attribute erweitern.

Ein Quiz besteht aus mehreren Zahlenfragen. Es gibt verschiedene Arten von Zahlenfragen, die im Folgenden genauer beschrieben werden.

Die Attribute sind im Diagramm nicht angegeben und können von Ihnen festgelegt werden.

Das Quiz dient hier lediglich zur Sammlung und Auswertung der Fragen, nicht aber zur eigentlichen Durchführung eines Quiz. Wenn Sie eine vollständige Implementierung haben, können Sie zum Ausprobieren die `main`-Methode der in [BeispielQuiz.java](#) enthaltenen Klasse ausführen.

Zahlenfrage:

Eine allgemeine Schnittstelle für Fragen, die eine Zahl als Antwort erwarten.

- `boolean istBeantwortet():` liefert `true` wenn die Methode `legeAntwortFest()` schon mindestens einmal ausgeführt wurde, ansonsten `false`
- `boolean istRichtigBeantwortet():` liefert `true` wenn die Frage schon beantwortet wurde und die Antwort richtig war, ansonsten `false`
- `void legeAntwortFest(int antwort):` legt die Antwort auf diese Frage fest
- `int liefereErreichtePunkte():` liefert die Anzahl der erreichten Punkte, abhängig von der Korrektheit der Antwort
- `String liefereFrage():` liefert den Fragetext der Aufgabe
- `int liefereMoeglichePunkte():` liefert die maximal mögliche Anzahl an erreichbaren Punkten

// Schnittstelle Zahlenfrage

```
interface Zahlenfrage {  
  
    boolean istBeantwortet();  
    boolean istRichtigBeantwortet();  
    void legeAntwortFest(final int antwort);  
    int liefereErreichtePunkte();  
    String liefereFrage();  
    int liefereMoeglichePunkte();  
}
```

AbstrakteZahlenfrage:

Eine abstrakte Klasse, die Teile der Schnittstelle Zahlenfrage implementiert.

- `AbstrakteZahlenfrage(String frage, int punkte)`: erzeugt eine neue Frage, mit der in `frage` übergebene Fragestellung und den in `punkte` übergebenen maximal möglichen Punkten
- `boolean istBeantwortet()`: entsprechend der Schnittstellenbeschreibung
- `void legeAntwortFest(int antwort)`: entsprechend der Schnittstellenbeschreibung
- `int liefereErreichtePunkte()`: liefert die maximal mögliche Anzahl an Punkte, wenn die Frage beantwortet wurde und wenn die Antwort richtig war, ansonsten 0
- `String liefereFrage()`: liefert die im Konstruktor übergebene Fragestellung
- `int liefereMoeglichePunkte()`: liefert die im Konstruktor übergebene maximal möglichen Punkten

```
// Klasse AbstrakteZahlenfrage
```

```
abstract class AbstrakteZahlenfrage implements Zahlenfrage{

    String frage;
    int maxPunkte;
    int erreichtePunkte;
    int antwort;
    boolean beantwortet;

    //Konstruktor
    public AbstrakteZahlenfrage(String frage, final int punkte) {
        this.frage = frage;
        this.maxPunkte = punkte;
    }

    public boolean istBeantwortet() {
        return this.beantwortet;
    }

    public void legeAntwortFest(final int antwort) {
        this.antwort = antwort;
        this.beantwortet = true;
    }

    public int liefereErreichtePunkte() {
        return this.erreichtePunkte;
    }

    public String liefereFrage() {
        return this.frage;
    }

    public int liefereMoeglichePunkte() {
        return this.maxPunkte;
    }

}
```

EinfacheZahlenfrage:

Eine Klasse, die überprüft ob die Antwort genau mit der richtigen Antwort übereinstimmt.

- `EinfacheZahlenfrage(String frage, int punkte, int richtigeAntwort)`: erzeugt

eine neue Frage, mit der in `frage` übergebene Fragestellung, den in `punkte` übergebenen maximal möglichen Punkten und der in `richtigeAntwort` übergebenen richtigen Antwort

- `boolean istRichtigBeantwortet()`: liefert `true`, wenn die Frage schon beantwortet wurde und der Wert der Antwort genau mit der im Konstruktor festgelegten Antwort übereinstimmt

```
// Klasse EinfacheZahlenfrage
```

```
class EinfacheZahlenfrage extends AbstrakteZahlenfrage {

    int richtigeAntwort;

    //Konstruktor
    public EinfacheZahlenfrage(final String frage, final int punkte, final int
richtigeAntwort) {
        super(frage, punkte);
        this.richtigeAntwort = richtigeAntwort;
    }

    //Implementiert abstrakte Klasse
    public boolean istRichtigBeantwortet() {
        if(this.beantwortet == true && this.antwort == this.richtigeAntwort) {
            this.erreichtePunkte += this.maxPunkte;
            return true;
        }
        else{
            return false;
        }
    }
}
```

SchaetzFrage:

Eine Klasse, die überprüft ob die Antwort innerhalb einer definierten Abweichung mit der richtigen Antwort übereinstimmt.

- `SchaetzFrage(String frage, int punkte, int richtigeAntwort, int prozentualeAbweichung)`: erzeugt eine neue Frage, mit der in `frage` übergebene Fragestellung, den in `punkte` übergebenen maximal möglichen Punkten, der in `richtigeAntwort` übergebenen richtigen Antwort und die Abweichung in Prozent, in der eine Antwort dennoch als richtig gilt
- `boolean istRichtigBeantwortet()`: liefert `true`, wenn die Frage schon beantwortet wurde und die gegebene Antwort sich innerhalb des Intervalls $[\text{richtigeAntwort} * (100 - \text{abweichung}) / 100, \text{richtigeAntwort} * (100 + \text{abweichung}) / 100]$ befindet.

```
// Klasse Schaetzfrage

class Schaetzfrage extends AbstrakteZahlenfrage {

    int richtigeAntwort;
    int prozentualeAbweichung;
    private final double HUNDERT_PROZENT = 100;
    //Konstruktor
    public Schaetzfrage(final String frage, final int punkte, final int
richtigeAntwort, final int prozentualeAbweichung) {
        super(frage, punkte);
        this.richtigeAntwort = richtigeAntwort;
        this.prozentualeAbweichung = prozentualeAbweichung;
    }

    //Implementiert abstrakte Klasse
    public boolean istRichtigBeantwortet() {
        double unterKante = (this.richtigeAntwort * (this.HUNDERT_PROZENT -
this.prozentualeAbweichung) / 100);
        double oberKante = (this.richtigeAntwort * (this.HUNDERT_PROZENT +
this.prozentualeAbweichung) / 100);
        if(this.antwort <= oberKante && this.antwort >= unterKante) {
            this.erreichtePunkte += this.maxPunkte;
            return true;
        }
        else{
            return false;
        }
    }
}
}
```

Quiz:

Diese Klasse repräsentiert ein Quiz. Ein Quiz bestehend aus mehreren Zahlenfragen. Es kann auf die im Quiz enthaltenen Fragen zugegriffen und die Gesamtpunktzahl berechnet werden. Die Durchführung eines Quizzes, also das Stellen der Fragen und das Einlesen der Antworten ist nicht Aufgabe dieser Klasse.

- Quiz (int fragenAnzahl): erzeugt ein neues Quiz, mit maximal fragenAnzahl Fragen
- boolean fuegeFrageHinzu(Zahlenfrage f): wenn schon die maximal mögliche Anzahl an Fragen im Quiz gespeichert wurde, wird false zurückgeliefert, ansonsten wird die Frage hinten ans Quiz angefügt und die Methode liefert true zurück; wird null übergeben, passiert nichts und es wird false zurückgeliefert
- int berechneErreichtePunkte(): liefert die Summe aller erreichten Punkte
- int berechneMaximalMoeglichePunkte(): liefert die Summe der maximal möglichen Punkte
- Zahlenfrage liefereFrage(int index): liefert die Frage an Position index; dabei ist die zuerst hinzugefügte Frage an Position 0 zu finden; gibt es zu dem Index keine Frage wird null zurückgeliefert
- int liefereFragenAnzahl(): liefert die Anzahl der im Quiz gespeicherten Fragen

```
// Klasse Quiz
```

```
public class Quiz {

    private int fragenAnzahl; //Anzahl der Gesamtfragen / Länge von fragenfeld
    private int fragenZaehler; //Zählervariable für Anzahl bereits
gespeicherter Fragen
    Zahlenfrage[] fragenFeld; //Array speichert die Fragen

    //Konstruktor
    public Quiz(final int fragenAnzahl){
        this.fragenAnzahl = fragenAnzahl;
        fragenFeld = new Zahlenfrage[fragenAnzahl]; //Arraylänge wird festgelegt
    }

    //Methoden
    public boolean fuegeFrageHinzu(final Zahlenfrage f){
        if(f == null) {
            return false;
        }else if(this.fragenZaehler < fragenAnzahl) {
            fragenFeld[this.fragenZaehler] = f; //aktuelle Frage
            //wird ins Array aufgenommen, in der Reihenfolge, in der sie gestellt wurde
            this.fragenZaehler++;
            return true;
        }else {
            return false;
        }
    }

    public int berechneErreichtePunkte() {
        int erreichtePunkte = 0;
        for(Zahlenfrage x : fragenFeld) {
            if(x != null) {
                erreichtePunkte += x.liefereErreichtePunkte(); //Schleife
            }
        }
        //addiert erreichte Punkte jeder Frage
        return erreichtePunkte;
    }

    public int berechneMaximalMoeglichePunkte() {
        int maxPunkte = 0;
        for(Zahlenfrage x : fragenFeld) {
            if(x != null) {
                maxPunkte += x.liefereMoeglichePunkte(); //Schleife addiert
            }
        }
        //max. mögl. Punkte jeder Frage
        return maxPunkte;
    }

    public Zahlenfrage liefereFrage(final int index) {
        if(index >= 0 && index <= fragenFeld.length) {
            return fragenFeld[index];
        }else {
            return null;
        }
    }

    public int liefereFragenAnzahl() {
        return this.fragenZaehler;
    }
}
```

Vergessen Sie nicht Ihr Programm hinreichend zu kommentieren.

Zum Ausprobieren des Spiels können Sie die `main`-Methode der in [BeispielQuiz.java](#) enthaltenen Klasse ausführen.

Testen Sie ihre Lösung, indem Sie den in [QuizTest.java](#) enthaltenen Testfall ausführen.