Hauptprüfung 2014/2015	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik	
	Teil: 2 (Pflichtbereich)	Aufgabe: 3 (5 Seiten)

Punkte

Lernspiel „Wörter bilden“

Für Grundschüler soll ein Lernprogramm entwickelt werden. Beim Starten eines neuen Spiels können Schüler zwischen zwei Spielvarianten wählen:

Typ 1: Lücken füllen (Ein fehlender Buchstabe in einem Wort muss ergänzt werden.)
Typ 2: Wortpuzzle (Aus den angezeigten Buchstaben muss ein Wort gebildet werden.)

Das Spiel ist für zwei Spieler konzipiert. Die Benutzeroberfläche ist unten dargestellt.

Neues Spiel starten:

Die Spieler tragen in die beiden Textfelder bei „Spieler 1“ bzw. „Spieler 2“ ihren Namen ein, wählen den Spieltyp (s. Auswahl Typ 1 oder 2) und betätigen die Taste „Starte neues Spiel“.

Bild 1: nach gedrückt: „Starte neues Spiel“

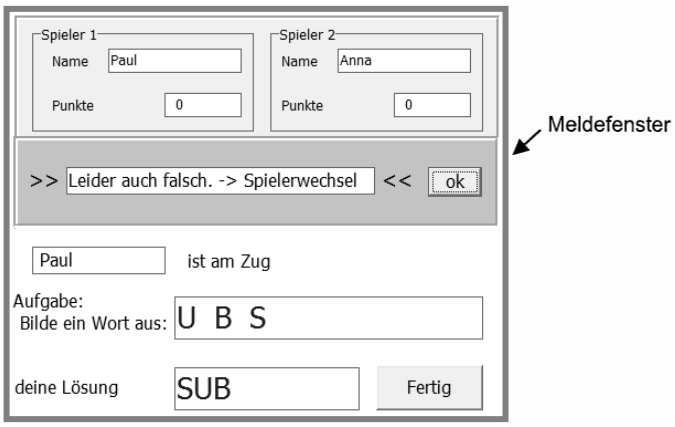


Bild 2: nach gedrückt: „Fertig“ (2. Lösungsversuch)

Spielverlauf:

Im Textfeld unterhalb der Starttaste (siehe Bild 1) wird angezeigt, welcher Spieler am Zug ist.
Im Textfeld hinter „Aufgabe …“ wird entweder ein Wort, bei dem ein Buchstabe fehlt (Typ 1), oder eine Folge von Buchstaben (Typ 2) ausgegeben.
Der Spieler, der am Zug ist, gibt in das Textfeld hinter „deine Lösung“ seine Lösung ein und drückt auf die Taste „Fertig“.
Ist seine Lösung richtig, bekommt er 5 Punkte. Wenn seine Lösung falsch ist, hat der Spieler einen zweiten Versuch. Bei diesem kann er jedoch nur noch einen Punkt erreichen.
Ein Spieler bleibt am Zug und bekommt solange neue Aufgaben gestellt, solange es ihm gelingt, beim ersten oder beim zweiten Versuch die richtige Lösung einzugeben.
Wenn die Antwort aber auch beim zweiten Lösungsversuch falsch ist, bekommt der andere Spieler die Möglichkeit, Punkte zu sammeln (siehe Bild 2, Spielerwechsel).
Nach dem Drücken auf die Taste „Fertig“ wird immer ein Meldefenster (siehe Bild 2) eingeblendet. Alle möglichen Meldungen (Texte) sind in Tabelle 1 bei Teilaufgabe 3.4 aufgelistet.
Zur Fortsetzung des Spiels muss der Spieler die Meldung mit der ok-Taste quittieren. Dadurch verschwindet das Meldefenster wieder (vergleiche Bild 1).

Ein anderer Entwickler hat schon ein Szenario modelliert (siehe Bild 3 auf der nächsten Seite) und ein unvollständiges Klassendiagramm (siehe Arbeitsblatt 1) angefertigt.

3.1.1 Definieren Sie den Begriff „Vererbung“. 1

3.1.2 Begründen Sie, weshalb die Klasse *Wort* (siehe Arbeitsblatt 1) als abstrakte Klasse definiert werden muss. 1

Hauptprüfung 2014/2015	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik	
	Teil: 2 (Pflichtbereich)	Aufgabe: 3 (5 Seiten)

Punkte

3.1.3 Abhängig vom ausgewählten Spieltyp werden beim Starten eines neuen Spiels entweder Objekte der Klasse *WortMitLuecke* oder Objekte der Klasse *WortPuzzle* erzeugt. 3

Ergänzen Sie im Klassendiagramm die Klassen *WortMitLuecke* und *WortPuzzle*. Alle benötigten Operationen müssen mit der Sichtbarkeit, der vollständigen Signatur und gegebenenfalls mit dem Datentyp des Rückgabewertes dargestellt werden.

Hinweise:
Beide Klassen benötigen die selben Attribute und Operationen wie die Klasse *Wort*.
Das Attribut *aWort* soll gleich beim Erzeugen eines Objekts durch den Konstruktor gesetzt werden und wird später nicht mehr verändert.

3.2 Bild 3 zeigt das Sequenzdiagramm für das Szenario: „Der Spieler drückt auf die Taste ***ok***. Da er schon zweimal eine falsche Lösung eingegeben hatte, kommt der andere Spieler zum Zug (→ Spielerwechsel)." 4

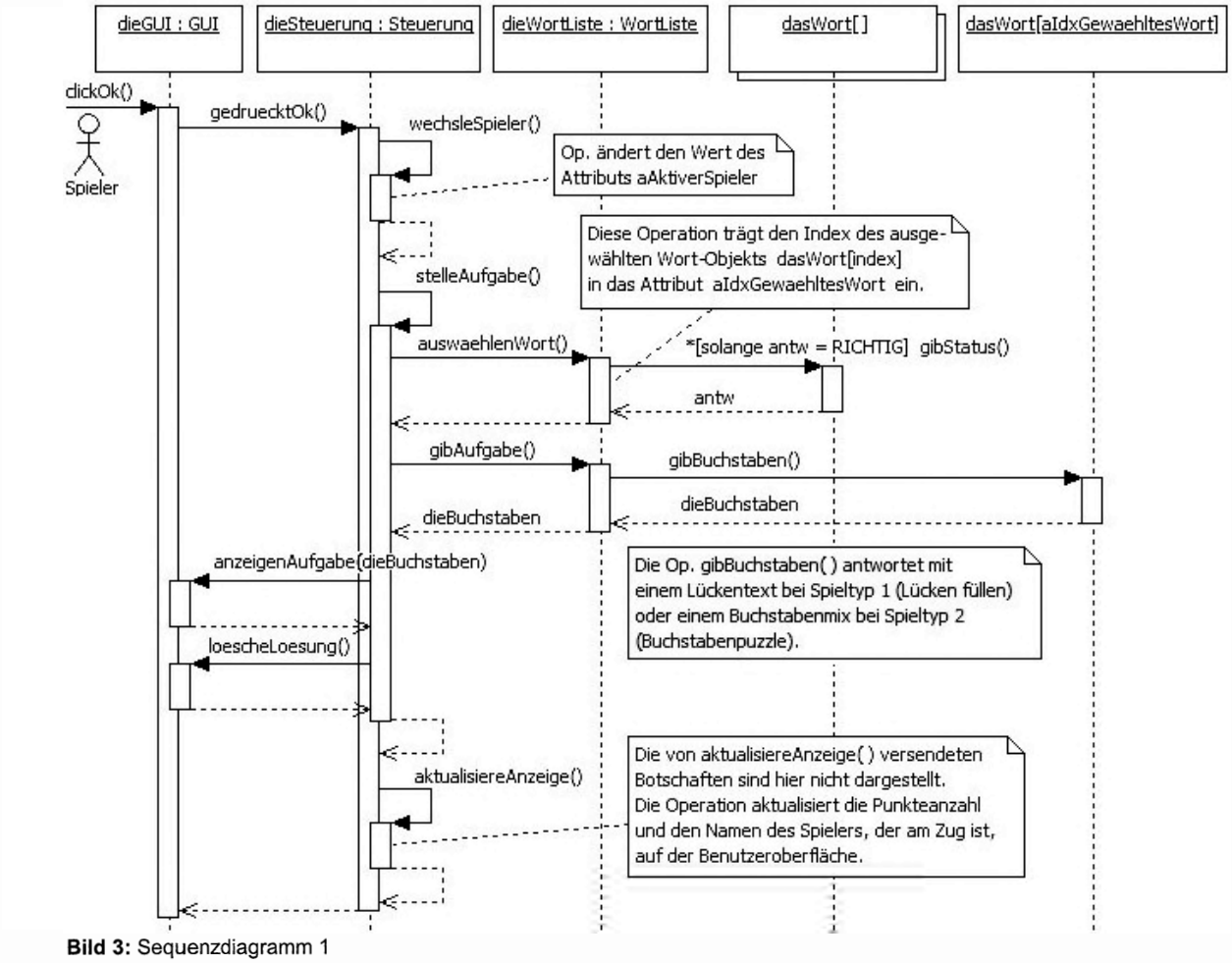


Bild 3: Sequenzdiagramm 1

Ergänzen Sie im Klassendiagramm auf dem Arbeitsblatt 1 alle Operationen und Assoziationen, die für das Sequenzdiagramm 1 (siehe Bild 3) benötigt werden.
Geben Sie bei Parametern und gegebenenfalls bei Operationen den Datentyp an.
Ergänzen Sie bei den Operationen die Sichtbarkeit. Die Sichtbarkeiten *public* und *protected* dürfen nur verwendet werden, wenn *private* nicht ausreicht.
Geben Sie bei Assoziationen die Richtung, die Rollennamen und die Kardinalitäten an, die sich aus dem Sequenzdiagramm 1 und der einleitenden Spielbeschreibung ableiten lassen.

Hauptprüfung 2014/2015	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik	
	Teil: 2 (Pflichtbereich)	Aufgabe: 3 (5 Seiten)

Punkte

3.3 Szenario modellieren 7

Vorgeschichte des Szenarios:

Der aktive Spieler (siehe Attribut *aAktiverSpieler* der Klasse *Steuerung*) hat schon einen erfolglosen Lösungsversuch unternommen. Nachdem er das Meldefenster quittiert hatte, hat er ein anderes Lösungswort eingegeben.

Beschreibung des Szenarios:

Der Spieler drückt auf die Taste ***Fertig***. Die Überprüfung, ob das eingegebene Lösungswort korrekt ist, wird vom *GUI*-Objekt an das Objekt von *Steuerung* delegiert. Dieses holt das vom Spieler eingegebene Wort mithilfe der Operation *leseLoesung( )* beim *GUI*-Objekt ab und sendet es an das Objekt von *Wortliste*.
Das *Wortliste*-Objekt ermittelt, ob das empfangene Wort mit dem Wort übereinstimmt, das in dem *Wort*-Objekt gespeichert ist, auf welches das Attribut *aIdxGewaehltesWort* verweist (vergleiche Bild 3, Notiz zur Operation *auswaehlenWort( )*).
Abhängig davon, ob das *Steuerung*-Objekt die Antwort *true* oder *false* erhält, wird entweder die Meldung: „Gut, das ist jetzt richtig! (1 Punkt)“ ausgegeben und zum Punktekonto des aktiven Spielers ein Punkt dazu addiert,
oder es wird die Meldung: „Leider auch falsch. ⇒ Spielerwechsel“ ausgegeben.
Das Szenario endet mit der Aktualisierung der Anzeige.

Erstellen Sie ein Sequenzdiagramm für das Szenario:
„Der Spieler drückt auf die Taste ***Fertig***. Es ist sein 2. Lösungsversuch.“
Das Hauptszenario: ‘Lösung richtig‘ und das Nebenszenario: ‘Lösung falsch‘ müssen dargestellt werden.
Die von der Operation *aktualisiereAnzeige( )* der Klasse *Steuerung* versendeten Botschaften können zur Vereinfachung der Aufgabe weggelassen werden.
Hinweis: Benutzen Sie zur Darstellung eine neue Seite im Querformat.

3.4 Entwickeln Sie ein Zustandsdiagramm für Objekte der Klasse *Steuerung*. 7
Das Zustandsdiagramm muss den auf Seite 1 beschriebenen Spielablauf modellieren.

Ergänzende Hinweise zum Spielablauf:

Die Tasten ***Fertig*** und ***ok*** müssen immer im Wechsel gedrückt werden.
Das Spiel kann nur durch Schließen des Fensters beendet werden (→ kein Endzustand).
Die Möglichkeit, den Spieltyp zu ändern, muss hier nicht berücksichtigt werden.
Welcher der beiden Spieler am Zug ist, wird durch den Wert des Attributs *aAktiverSpieler* festgelegt.

Beispiel für einen möglichen Spielablauf (Erläuterung der Abkürzungen s. Tabelle 1):

StelleAufg → Eingabe: richtige Lösung, Taste ***Fertig*** → Mld_1 → Taste ***ok*** →
StelleAufg → Eingabe: fehlerhafte Lösung, Taste ***Fertig*** → Mld_2 → Taste ***ok*** →
Eingabe: richtige Lösung, Taste ***Fertig*** → Mld_3 → Taste ***ok*** →
StelleAufg → Eingabe: fehlerhafte Lösung, Taste ***Fertig*** → Mld_2 → Taste ***ok*** →
Eingabe: fehlerhafte Lösung, Taste ***Fertig*** → Mld_4 → Taste ***ok*** →
WechsleSpieler, StelleAufg → …

Somit können nur die hier aufgelisteten Ereignisse und Wächterbedingungen auftreten:

- gedruecktFertig( ) [Lösung richtig]
- gedruecktFertig( ) [Lösung falsch]
- gedruecktOk( )

Fortsetzung der Aufgabe auf der nächsten Seite

Hauptprüfung 2014/2015	Berufliches Gymnasium (TG)	
1.5.2	Informationstechnik	
	Teil: 2 (Pflichtbereich)	Aufgabe: 3 (5 Seiten)

Punkte

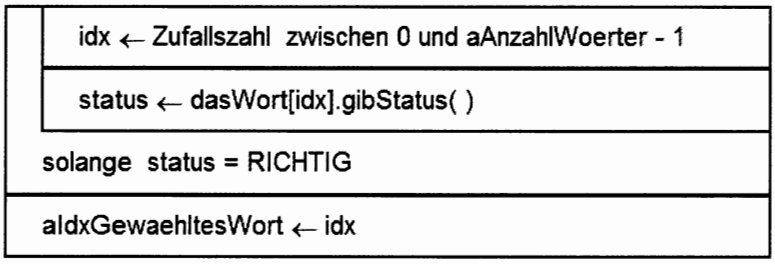
Nur die in Tabelle 1 (s. unten) aufgelisteten Aktionen müssen berücksichtigt werden. Sie können die in Spalte 1 eingetragenen Abkürzungen im Zustandsdiagramm verwenden.

Abkürzung	Aktion
Mld_1	*dieGUI.anzeigenMeldung("*Prima, das ist richtig! (5 Punkte)*")*
Mld_2	*dieGUI.anzeigenMeldung("*Falsch, aber du hast noch eine Chance!*")*
Mld_3	*dieGUI.anzeigenMeldung("*Gut, das ist jetzt richtig! (1 Punkt)*")*
Mld_4	*dieGUI.anzeigenMeldung("*Leider auch falsch. ⇒ Spielerwechsel*")*
Add_1Pkt	*derSpieler[aAktiverSpieler].addPunkte(1)*
Add_5Pkt	*derSpieler[aAktiverSpieler].addPunkte(5)*
WechsleSp	*wechsleSpieler( )* // Ändert den Wert des Attributs *aAktiverSpieler*.
StelleAufg	*stelleAufgabe( )* // Eine neue Aufgabe wird ausgewählt und angezeigt.

Tabelle 1: Meldungen und andere Aktionen

3.5 Für die Operation *auswaehlenWort( )* der Klasse *WortListe* liegt der unten dargestellte Algorithmenentwurf vor.

```
Konstante:  RICHTIG = 1
lokale Variable:
  idx: GZ
  status: GZ
```



3.5.1 Algorithmus analysieren 2

a) Erläutern Sie die Arbeitsweise dieses Algorithmus.

b) Untersuchen Sie, was passiert, wenn alle Aufgaben gelöst sind, und beschreiben Sie die zu erwartende Reaktion des Lernspielprogramms.
Hinweise: Bei einer korrekt gelösten Aufgabe antwortete das entsprechende *Wort*-Objekt auf die Botschaft *gibStatus( )* mit *RICHTIG (= 1)*. Bei bisher nicht gelösten Aufgaben antwortet die Botschaft mit *FALSCH* (= -1) oder *NIE* (= 0).

3.5.2 Entwickeln Sie für die bei 3.5.1 festgestellte Problematik einen Verbesserungsvorschlag und stellen Sie den erweiterten Algorithmus für die Operation *auswaehlenWort( )* in einem Struktogramm dar. 5
Anmerkung: Auch wenn alle Aufgaben schon einmal gelöst wurden, soll es möglich sein, weiter zu spielen.

30

MINISTERIUM FÜR KULTUS, JUGEND UND SPORT BADEN-WÜRTTEMBERG

Zu- und Vorname:		Schulnummer	Schülernummer
Hauptprüfung 2014/2015			
1.5.2	Informationstechnik		

Hauptprüfung 2014/2015			Schulnummer	Schülernummer
1.5.2	Informationstechnik			
Arbeitsblatt	Teil: 2	Aufgabe: 3		

Arbeitsblatt 1: Klassendiagramm