

# KORREKTURBOGEN

Vorlesung Programmieren

Probe-Programmmentwurf 'DartsCounter'



**Hilfsmittel:** Vorlesungsfolien, Übungsblätter + Aufgaben

**Bearbeitungszeit:**  
90 Minuten

## Bewertung:

Kriterium			Mögl. Punkte	Erreichte Punkte
Test	t1	DartsCounter starten Programm startet ohne Fehler und beendet nicht sofort wieder (Eingabe erwartet)	5	
	t2	Spielen Spielernamen und Restpunktzahl sind zu sehen, Hinweis für Eingabe des Wurfs vorhanden	5	
	t3	Erfolgreich spielen 9-Darter eingeben (2x 'T20 T20 T20' + 'T20 T19 D12' für denselben Spieler) führt zu Erfolgsmeldung und Rest-Punktestände sind korrekt (501/321/141)	5	
	t4	Zu lange spielen nur 1er Aufnahmen eingeben, Spiel bricht mit Information nach Wurfdurchgang 10 ab	5	
	t5	Treffer 'daneben' Eingabe einer Aufnahme mit nur 2 getroffenen Darts 'T20 T20' sollte 120 Punkte und keinen Fehler ergeben	5	
a	a1	Klasse Field korrekt angelegt	1	
	a2	Alle 3 Attribute korrekt angelegt int value / String label / boolean doubleField (1) + private (1)	2	
	a3	Konstruktor korrekt angelegt alle 3 Attribute als Argumente	1	
	a4	Getter-Methoden vorhanden Getter-Methoden alle vorhanden	1	
b	b1	Klasse Board korrekt angelegt Angelegt (1), geeignete Datenstruktur für Felder als Instanz-Variable angelegt (1)	2	
	b2	Felder erzeugt Durch Konstruktor erzeugt (1), alle vorhanden, korrekte Labels und Werte, Doppelfeld-Flag (insg. 3)	4	
	b3	Methode parseField korrekt angelegt (1), alle Felder durchlaufen & verglichen (1), case-insensitive (1), null-Rückgabe wenn nichts gefunden (1)	4	
c	c1	Klasse Visit korrekt angelegt	1	
	c2	Instanz-Variable für Felder vorhanden private Field[] fields	1	
	c3	Konstruktor Übergabe von Field[] (1), Prüfung der Länge + Exception wenn zu lang (2)	3	
	c4	Methoden getFields/getValue/getLastField Richtig definiert (insg. 1), getFields liefert Instanz-Variable (1), getValue liefert aufsummierten Wert aller Felder (2), getLastField liefert letztes Field	5	
d	d1	Klasse Player korrekt angelegt	1	
	d2	Instanz-Variablen + Getter String name, int countDartsThrown, Visit[] visit (insg. 1), alle private (1), Getter (1)	3	
	d3	Methode getRemainingPoints Korrekt 501 minus aktuelle Punkte berechnet und zurückgegeben	2	
	d4	Methode addVisit Prüfung ob Restpunkte kleiner 0 (1), wenn nötig Argument in Array speichern (1), in jedem Fall countDartsThrown erhöhen (1), korrekte Rückgabe (1)	4	
e	e1	Klasse Game korrekt angelegt	1	
	e2	Konstruktor + Instanz-Variablen Konstruktor mit Argumenten wie in main-Vorgabe (1), Speicherung des Boards und der Player in Instanz-Variablen (1)	2	
	e3	Spielablauf 1: prinzipielle Spielstruktur Methode start korrekt angelegt (1), Es werden immer nacheinander alle Spieler durchlaufen (2), aktueller Spieler + Restpunkte ausgegeben (2), die Eingabe abgefragt (1), Score des Wurfs danach angezeigt (1)	7	
	e4	Spielablauf 2: Einlesen der Eingabe Eingabe aufgesplittet in die 3 Würfe (1), jeder Wurf wird zu einer Field-Instanz geparkt (2), daraus ein Visit-Objekt erzeugt (2)	5	
	e5	Spielablauf 3: Abbruchkriterien Prüfung nach jedem 'Visit' ob 0 Punkte Rest (2), Prüfung nachdem alle Spieler dran waren ob 10 Durchgänge (1), Ausgabe des Siegers bzw. Info zu Spielende (2)	5	

f	f1	Schreibender Dateizugriff Datei wird zum Schreiben geöffnet (1), mit append (1), Ressource wird sicher geschlossen (1)	3	
	f2	Zeile einfügen Zeile wird eingefügt (1), Name und Wurfszahl erwähnt (1)	2	
g	g1	Double-Out Bei Restpunktzahl 0 Prüfung auf Doppelfeld-Flag (isDoubleField) des letzten Feldes (getLastField)	2	
	g2	Restpunktzahl 1 Visit wird abgelehnt, wenn Restpunktzahl 1 Punkt wäre	2	
	g3	Anzahl geworfener Darts countDartsThrown wird weiterhin erhöht, auch wenn kein Double-Out oder Restpunktzahl 1	1	
h	h1	Prüfung der Restpunktzahl Prüfung, ob die Restpunktzahl 170 oder weniger beträgt	1	
	h2	Lesender Dateizugriff + Einlesen der Zeilen Bereitgestellte Datei (1) wird zum Lesen geöffnet (1), Ressource wird sicher geschlossen (1), Zeilen werden entweder in Datenstruktur eingelesen oder bis zur korrekten Zeile jedesmal gelesen (3)	6	
	h3	Ausgabe des Checkouts Der korrekte Checkout-Weg der aktuellen Restpunktzahl wird ausgegeben	3	
Summe			100	

#### Notenskala:

>= 94 (100%): 1,0	93 (99%): 1,0	92 (98%): 1,1	91 (97%): 1,1	90 (96%): 1,2	89 (95%): 1,3
88 (94%): 1,3	87 (93%): 1,4	86 (92%): 1,5	85 (91%): 1,5	84 (90%): 1,6	83 (89%): 1,7
82 (88%): 1,7	81 (87%): 1,8	80 (86%): 1,8	79 (85%): 1,9	78 (83%): 2,0	77 (82%): 2,0
76 (81%): 2,1	75 (80%): 2,2	74 (79%): 2,2	73 (78%): 2,3	72 (77%): 2,4	71 (76%): 2,4
70 (75%): 2,5	69 (74%): 2,5	68 (73%): 2,6	67 (72%): 2,7	66 (71%): 2,7	65 (70%): 2,8
64 (69%): 2,9	63 (68%): 2,9	62 (66%): 3,0	61 (65%): 3,1	60 (64%): 3,1	59 (63%): 3,2
58 (62%): 3,2	57 (61%): 3,3	56 (60%): 3,4	55 (59%): 3,4	54 (58%): 3,5	53 (57%): 3,6
52 (56%): 3,6	51 (55%): 3,7	50 (54%): 3,8	49 (53%): 3,8	48 (52%): 3,9	47 (50%): 4,0
46 (49%): 4,0	45 (48%): 4,1	44 (47%): 4,1	43 (46%): 4,2	42 (45%): 4,3	41 (44%): 4,3
40 (43%): 4,4	39 (42%): 4,5	38 (41%): 4,5	37 (40%): 4,6	36 (39%): 4,7	35 (38%): 4,7
34 (37%): 4,8	33 (36%): 4,8	32 (35%): 4,9	<= 31 (33%): 5,0		