# Übung: Dartspiel 1

Hinweis: Beachten Sie zur Bearbeitung der nachfolgenden Aufgabenstellungen das Informationsmaterial

*L2\_2\_2 Info kopfgesteuerte Schleife.docx*

*L2\_2\_2 Schuelerdaten Dartspiel1.py*

**(I) Problemstellung**

Sie wurden beauftragt für den Dartsportverein „Wilder Dart“ ein Programm zu schreiben, das den Spielern die Zählarbeit beim Spiel X01 abnimmt. Im Vorfeld wurde eine UML-Klassendiagramm entworfen (vgl. (4) UML-Klassendiagramm) und mit der Implementierung der Klasse Dartspiel begonnen.

1. Öffnen Sie das Programm *L2\_2\_2 Schuelerdaten Dartspiel1.py*
2. Schreiben Sie ein Programm, mit dem das Dartspiel X01 gespielt werden kann. Speichern Sie Ihre Lösung in Ihrem Ergebnisordner unter dem Namen *L2\_2\_2 Loesung Dartspiel 1.py.* Beachten Sie bei der Programmierung der Methode *spiel\_starten()* die folgenden Punkte:
   1. Es soll dem Nutzer zuerst der Startwert angezeigt werden (bspw. 101 oder 301).
   2. Der Nutzer hat die Möglichkeit, seine geworfenen Punkte nach jedem Wurf einzugeben. Die geworfenen Punkte werden nach und nach vom Startwert/Restwert abgezogen.
   3. Dem Nutzer soll nach jedem Wurf der aktuelle Punktestand angezeigt werden.
   4. Der Nutzer muss genau 0 Punkte treffen, d. h. wirft er eine höhere Punktzahl als sein aktueller Punktestand, zählen die Punkte des Wurfes nicht und der Punktestand bleibt unverändert.
   5. Das Spiel endet, wenn der Punktestand genau bei 0 Punkte ist. Am Ende soll dem Nutzer die Anzahl der benötigten Würfe angezeigt werden.
3. Testen Sie Ihr Programm mit der entsprechenden Codierung im Hauptprogramm.

**(II) Problemanalyse**

1. Welche Ausgabedaten will man erhalten?

Aktuelle Punkte, Anzahl der Würfe, Punkte Wurf

1. Welche Daten werden zur Bearbeitung benötigt?

-

1. Welche Eigenschaften haben die Eingabe-, Verarbeitungs- und Ausgabedaten? (**Variablenliste**)

|  |  |  |
| --- | --- | --- |
| **Bedeutung** | **Typ** | **Variable** |
| Aktuelle Punkte | Ganzzahl | pkt |
| Anzahl der Würfe | Ganzzahl | wuerfe |
| Punkte Wurf | Ganzzahl | pkt\_wurf |

1. UML-Diagramm:



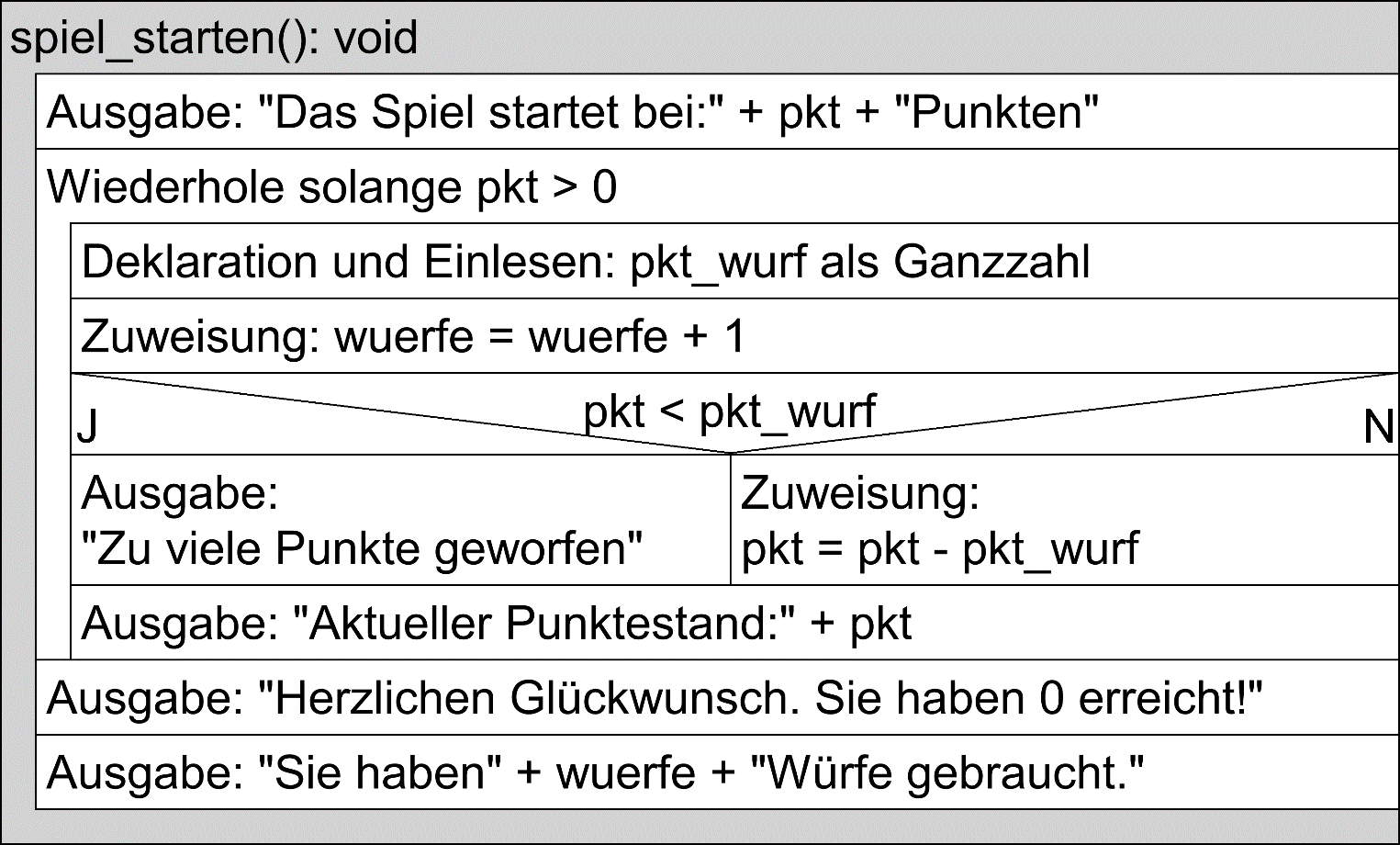
1. So soll die Bildschirmausgabe des Programms aussehen (Beispieldaten):

|  |
| --- |
| Das Spiel startet bei: 101 Punkten  Punkte: **60**  Aktueller Punktestand: 41  Punkte: **40**  Aktueller Punktestand: 1  Punkte: **2**  Zu viele Punkte geworfen  Aktueller Punktestand: 1  Punkte: **1**  Aktueller Punktestand: 0  Herzlichen Glückwunsch. Sie haben 0 erreicht!  Sie haben 4 Würfe gebraucht |

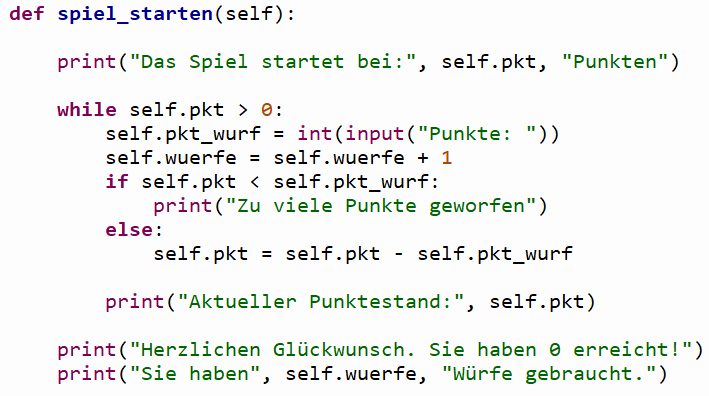
1. Verarbeitung

|  |
| --- |
| Punkteverarbeitung:  WENN pkt < pkt\_wurf  DANN Ausgabe zu viele Punkte geworfen  SONST pkt = pkt – pkt\_wurf |

**(III) Struktogramm**



**(IV) Programmcode (Python-Code)**



**Datei:** L2\_2\_2\_Loesung\_Dartspiel1.py