

Aufgabenstellung 1: Kaffeeautomat

A

Erstelle einen Kaffeeautomaten der in der Lage ist unterschiedlichen Kaffee zu servieren. Für jeden Automaten soll es nur eine einzige Instanz geben, die es ermöglicht Kaffee zu bestellen und mit Zucker, extra Zucker, Milch, extra Milch, Milchschaum. Der Kaffeeautomat kann auch Kakao und Tee servieren. Der Automat soll die Bestellung entgegennehmen, den Preis ausgeben und nach Bezahlung einen Kaffee mit allen gewünschten Zutaten servieren. Der Preis setzt aus dem Basiskaffee z.B. Melange, Espresso, Häferlkaffee, Filterkaffee, ... und den extra Zutaten zusammen.

Benutzerinteraktion einbauen:

Erstelle den Kaffeeautomaten als Logik, welche in einer Consolenanwendung sowie in einer GUI verwendet werden kann. Wähle selbst eine Oberfläche deiner Wahl und erstelle einen Kaffeeautomaten.

Aufgabenstellung 2: Sortierer

C

Entwirf einen Sortierer, der alles (alle Datentypen: Zahlen, Zeichenketten, Buchstaben) sortieren kann. Die Daten können in einem Feld, in einer Datei oder einer Liste gespeichert sein. Beim Benutzen des Sortierers gibt es die Möglichkeit zwischen 5 unterschiedlichen Sortieralgorithmen zu unterscheiden. Wähle 5 Algorithmen deiner Wahl und implementiere diese so, dass eine Zeitmessung erfolgt und das Ergebnis dieser zur Verfügung gestellt wird.

Stelle die Sortieralgorithmen als Logik für andere Programme zur Verfügung.

Erweiterung des Sortierers

- Ermögliche nach Attributen einer Klasse zu sortieren:
 - Erklärung: Beispielsweise Klasse Person besteht aus Id, Vorname, Nachname, PLZ, Ort, usw. Ermögliche es, dass die Personen nach beliebigen Elementen der Klasse Person sortiert werden können – nach Name oder nach PLZ, ...
- Erstelle eine grafische Darstellung der unsortierten Listen und vergleiche die Zeitmessungen je Sortieralgorithmus (beispielsweise Vergleich via Chart).

Aufgabenstellung 3: Prozessor

C

Gegeben ist der Prozessor eines Computers, dieser verwaltet 3 Zustände: bereit, aktiv, blockiert und diverse Zustandsübergänge. Desweiteren kann ein Prozess erzeugt und zerstört werden. Bilde einen Prozessor mit allen Prozesszuständen und den Übergängen ab (Details sieh Grafik). Erstelle Aufgabenstellungen (einfache Mathematische Aufgaben (Tasks: Mul, Div, Add, Sub, ...)) die der Prozessor der Reihe nach lösen soll und Ausführen soll. Ein Task kann beliebig lang sein und aus mehreren Teiltasks bestehen. Der Prozessor steht immer nur für maximal drei Teiltasks zur Verfügung, anschließend erfolgt ein Timeout und der Task landet mit den verbleibenden Teiltasks in der Warteschleife bis der Prozessor wieder frei ist. Für den Prozessor ist wichtig, dass er einen Task einfach „ausführen“ kann, das bedeutet: der Task selbst beinhaltet eine Methode zum Lösen der Aufgabenstellung mit einer Ausgabe, sodass ersichtlich ist, dass dieser Task gelöst wurde. Das gesamte Programm ist zu instrumentalisieren (mit Ausgaben zu versehen), sodass der Ablauf gut nachvollziehbar ist.

Aufgabenstellung 4: Pizzeria

C

Erstelle für eine Pizzeria mit mehreren Standorten eine Applikation die es ermöglicht je Standort unterschiedliche lokale Spezialpizzas zu backen. In manchen Orten mögen die Kunden besonders viel Käse, in anderen Orten lieben die Kunden eine knusprige Rinde, in der nächsten Ortschaft ist es Standard eine Pizza mit Knoblauch zu bestreichen vor dem Servieren und als Sonderspezialität gibt es für Vegetarier eine eigene Pizzeria mit nur Veganen und Vegetarischen Pizzas. Für die besonders anspruchsvollen Kunden gibt es Gluten freien Pizzateig als Auswahlmöglichkeit. Die Pizzasauce kann auch jeweils unterschieden werden zwischen Normal, extra viel Sauce, nur Tomatenmark oder keine Sauce.

Unserem Unternehmen ist es wichtig, dass jede Pizzeria sich durch ein Spezialmerkmal am Markt positioniert. Jede Pizzeria hat ein Standardangebot an Pizzas und die Möglichkeit diese Standardpizzen speziell mit unterschiedlichen Zutaten zu gestalten. Der Preis je Pizza wird berechnet durch die Grundkomponenten der Basispizza und den jeweils einzelnen dazu kommenden Zutaten. Wenn eine Pizza bestellt wird soll jeweils der Preis der Pizza ermittelt werden und anschließend der gesamte Vorgang dokumentiert werden. Aus welchen Bestandteilen setzt sich die Pizza zusammen, wie lange wird sie gebacken, wird sie geschnitten oder im Ganzen serviert, wie viel hat der Kunde für die Pizza zu zahlen.

Aufgabenstellung 5: Schach

B

Recherchiere selbst die genauen Spielregeln und implementiere diese.

Hinweis:

Erstelle eine Logik für das Spielen von Schach.

Wechsle zwischen zwei Spielern automatisch hin und her.

Teste alle möglichen Spielzüge mit Hilfe von Unit Tests

Ermögliche diese Logik in einer Consolenanwendung sowie in einer grafischen Benutzeroberfläche einzubinden.

Aufgabenstellung 6: Schiffe versenken

B

Recherchiere selbst die genauen Spielregeln und implementiere diese.

Hinweis:

Erstelle eine Logik für das Spielen von Schiffe versenken.

Wechsle zwischen zwei Spielern automatisch hin und her.

Teste alle möglichen Spielzüge mit Hilfe von Unit Tests

Ermögliche diese Logik in einer Consolenanwendung sowie in einer grafischen Benutzeroberfläche einzubinden.

Aufgabenstellung 7: 4 Gewinn

B

Recherchiere selbst die genauen Spielregeln und implementiere diese.

Hinweis:

Erstelle eine Logik für das Spielen von 4 Gewinn.

Wechsle zwischen zwei Spielern automatisch hin und her.

Teste alle möglichen Spielzüge mit Hilfe von Unit Tests

Ermögliche diese Logik in einer Consolenanwendung sowie in einer grafischen Benutzeroberfläche einzubinden.

Aufgabenstellung 8: Strategiespiel-> Fortbewegung

C

Erstelle ein rundenbasiertes Strategiespiel die Bewegungskosten der Figuren zu berechnen. Jeder Figur stehen in einer Spielrunde eine bestimmte Zahl von Bewegungspunkten zur Verfügung. Ist dieses Punktekonto aufgebraucht, kann die Figur in der entsprechenden Runde nicht mehr bewegt werden.

Das Spielfeld ist in Felder aufgeteilt. Für jedes Feld auf das sich eine Figur bewegt muss eine bestimmte Anzahl von Bewegungspunkten bezahlt werden. Die Kosten für die Bewegung auf ein bestimmtes Feld werden vom Bewegungskonto abgebucht sobald sich die Figur auf dieses Feld bewegt. Übersteigen die Bewegungskosten für ein bestimmtes Feld das Bewegungskonto der Figur, kann sie sich auf dieses Feld nicht bewegen.

Jedem Spielfeld ist ein bestimmter Geländetyp (Gras, Geröll, Fels, Wasser, Schlamm, Waldboden, usw.). Für die unterschiedlichen Geländetypen unterscheiden wir unterschiedliche Geländekosten.

Für eine Spielfigur gibt es unterschiedliche Arten der Fortbewegung (Gehen, Laufen, Reiten, Fliegen, Levitieren (freischwebend), usw.). Die Kosten der Fortbewegung für ein bestimmtes Feld berechnen sich als Kombination des Geländetyps und der Art der Fortbewegung dieser Spielfigur.

Aufgabenstellung 9: Schach

C

Erstelle ein rundenbasiertes Strategiespiel um den Kampf zwischen 2 Figuren zu simulieren. Jede Figur besitzt einen Angriffsverhalten und ein Verteidigungsverhalten.

Das Angriffsverhalten einer Figur setzt sich aus 2 Werten zusammen: der Reichweite und dem Schaden. Die Reichweite beschreibt wie viele Felder eine Figur maximal entfernt sein kann, damit der Angriff erfolgreich ist. Der Schaden beschreibt die Anzahl der Punkte die die Figur von seinen Lebenspunkten abziehen muss. Fallen die Lebenspunkte der Figur auf oder unter 0 wird die Figur aus dem Spiel genommen.

Das Verteidigungsverhalten einer Figur besteht aus 2 Werten: dem Rüstungswert und dem Ausweichwert. Der Rüstungswert beschreibt die Art von Rüstung, die die Figur trägt. Wird im Rahmen eines Angriffs gegen die Figur Schaden verursacht, wird der Rüstungswert vom Schaden abgezogen bevor er von den Lebenspunkten abgezogen wird. Der Ausweichwert wird von der Reichweite des Angriffs abgezogen. Liegt die Figur nicht mehr innerhalb der möglichen Reichweite des Angriffs schlägt dieser fehl.

Angriff und Verteidigung werden über Würfel simuliert. Soll ein Angriff durchgeführt werden Würfel geworfen. Es gibt 4 Arten von Würfel: rot, gelb, blau und grün. Jede Seite eines Würfels zeigt 2 Werte. Einen Schadenswert und einen Reichweitewert.

Seiten eines roten Würfels:

- Reichweite 0, Schadenswert 0
- Reichweite 0, Schadenswert 0
- Reichweite 1, Schadenswert 1
- Reichweite 2, Schadenswert 1
- Reichweite 3, Schadenswert 1
- Reichweite 4, Schadenswert 0

Seiten eines blauen Würfels:

- Reichweite 0, Schadenswert 0
- Reichweite 6, Schadenswert 0
- Reichweite 3, Schadenswert 1
- Reichweite 5, Schadenswert 1
- Reichweite 4, Schadenswert 1
- Reichweite 4, Schadenswert 1

Seiten eines gelben Würfels:

- Reichweite 0, Schadenswert 0
- Reichweite 0, Schadenswert 0
- Reichweite 1, Schadenswert 1
- Reichweite 1, Schadenswert 1
- Reichweite 1, Schadenswert 2
- Reichweite 2, Schadenswert 1

Seiten eines grünen Würfels:

- Reichweite 0, Schadenswert 0
- Reichweite 4, Schadenswert 0
- Reichweite 5, Schadenswert 0
- Reichweite 7, Schadenswert 0
- Reichweite 5, Schadenswert 1
- Reichweite 8, Schadenswert 1

Einer Figur können mehrere unterschiedliche Würfel für den Angriff zugeordnet werden. Führt eine Figur einen Angriff durch dann wird für jeden Würfel einen Würfelwurf simuliert. Der Schadenswert des Angriffs ergibt sich aus der Summe aller geworfenen Schadenssymbole. Der Reichweitewert des Angriffs entspricht dem Maximalen geworfenen Reichweitewert aller Würfel.