

04 – Verständnisfragen zur selbstständigen Beantwortung

- (1) Warum haben Zustandsdiagramme eine besondere Bedeutung bei der Kommunikation zwischen Programmierern und Designern?
- (2) Nennen Sie sechs Gründe, die für den Einsatz einer Zustandsmaschine sprechen.
- (3) Erläutern Sie den Vorteil von Enumeratoren im Vergleich zu einem `int` bzw. einem `String` (Zeichenkette) bei der Implementierung einer Zustandsmaschine in Java.
- (4) Was ist eine Fuzzy-Finite-State-Machine? Welche Motivation liegt dieser Form einer Zustandsmaschine zu Grunde?



04 – Aufgabensammlung

- (1) Verbessern Sie durch Umstrukturierung und Auslagerung von Programmteilen in Funktionen die Eigenschaften Übersichtlichkeit, Erweiterbarkeit und Wartbarkeit des Programms **JavaGotchi**.

```
public class JavaGotchi {
    enum Zustand {HUNGER, VERHUNGERT, ESSEN, GLÜCKLICH};

    public static void main(String[] args) {
        Zustand zustand = Zustand.GLÜCKLICH;
        int hunger = 0;
        boolean spielLaeuft = true;      // Kleiner Gameloop für unser JavaGotchi
        while (spielLaeuft) {
            switch (zustand) {
                case HUNGER:
                    System.out.println(":/ ... hunger");
                    if (Math.random() < 0.25) // in 25% der Fälle finden wir etwas Essbares
                        zustand = Zustand.ESSEN;
                    break;
                case VERHUNGERT:
                    System.out.println("x(");
                    spielLaeuft = false;
                    break;
                case ESSEN:
                    System.out.println(":# ... essen");
                    hunger = 0;
                    zustand = Zustand.GLÜCKLICH;
                    break;
                case GLÜCKLICH:
                    System.out.println(":)");
            }

            if (zustand != Zustand.ESSEN) {
                hunger += (int)(Math.random()*5);
                if (hunger > 15) zustand = Zustand.HUNGER;      // :/
                if (hunger > 30) zustand = Zustand.VERHUNGERT;  // x(
            }
        }
    }
}
```

