

T01 // Immigration

COR, Bridge, Observer

Die Einwanderungskontrolle an einem Flughafen hat vier Schalter (i) Diplomat, (ii) EU Bürger, (iii) Nicht-EU Bürger und (iv) EasyPass mit jeweils einer definierten Zuständigkeit. Standardmäßig sind die Schalter (i) bis (iii) geöffnet. Schalter (iv) ist geschlossen. Alle zwei Sekunden trifft ein neuer Fluggast ein. Der Typ Diplomat, EU oder Nicht-EU des Fluggastes wird mit einem geeigneten Zufallsgenerator bestimmt. Die Bearbeitung an einem Schalter dauert zwischen 1-3 Sekunden (zufällig bestimmt). Ein Fluggast, der nicht unmittelbar durch einen Schalter bedient werden kann, wird in eine zentrale Warteschlange eingereiht. Sobald der relevante Schalter frei ist (Bearbeitung abgeschlossen), wird der erste Fluggast aus der korrespondierenden Warteschlange „entnommen“ und die Bearbeitung durchgeführt. Der Schalter (iv) beobachtet kontinuierlich die Warteschlange für die Schalter (i) Diplomat, (ii) EU Bürger und (iii) Nicht-EU Bürger. Sobald sich fünf Fluggäste in einer Warteschlange befinden, werden die ersten drei Fluggäste aus der Warteschlange „entnommen“, der Schalter (iv) geöffnet, jeder Fluggast innerhalb von einer Sekunde durch Schalter (iv) bearbeitet und dann der Schalter (iv) wieder geschlossen. Bei Schalter (iv) können zwei Varianten R1 und R2 von Lesegeräten für den Reisepass zum Einsatz kommen.

T02 // Zelle

Composite, Observer, Visitor

Eine (genetisch veränderte Zelle) besteht aus zwei Subzellen. Eine Subzelle hat wiederum zwei Subzellen. Das Immunsystem beobachtet kontinuierlich jede Subzelle. Sobald ein Virus in eine Subzelle eindringt, wird das Immunsystem automatisch informiert. In Abhängigkeit von dem Typ (i) V1 oder (ii) V2 des Virus steht jeweils ein definierter Zerstörungsmechanismus zur Verfügung. Aus Aspekten der Vereinfachung wird ein Virus als Matrix mit 1000 Zeilen und 25 Spalten mit Zeichen aus dem Pool [a-z,0,9] dargestellt. V1 hat mindestens 70% der Zeichen aus dem Pool [a-z] und V2 mindestens 70% der Zeichen aus dem Pool [0,9]. Der geeignete Zerstörungsmechanismus besucht die infizierte Subzelle und entfernt das Virus. Des Weiteren befindet sich vertikal, diagonal oder horizontal (zufällig bestimmt) die Zeichenkette ebola für V2 und lassafever für V1.

T03 // Intelligenter Zugbegleiter

Builder, Bridge, Adapter

Ein Zug hat einen Waggon. Dieser Waggon hat auf der linken und rechten Seite je 25 Sitzplätze – getrennt durch einen Gang. Das elektronische Ticket besteht aus den Informationen Reisender (name und fingerprint), Datum, Klasse sowie Strecke von und nach. Ein Builder erstellt das Ticket und registriert es in einem zentralen Repository. Zusätzlich wird das Ticket dem Reisenden auf das MPhone gesandt. Das MPhone ist mit einem RFID-Chip ausgestattet. Der RFID-Chip existiert in den Varianten EU und US. Der intelligente Zugbegleiter besucht sukzessive die belegten Sitzplätze und überprüft die Tickets anhand der registrierten Tickets im zentralen Repository. Standardmäßig liest der intelligente Zugbegleiter den RFID-Chip der Variante EU. Für das Auslesen des RFID-Chips der Variante US nutzt der intelligente Zugbegleiter einen Adapter.

Eine menschliche Niere hat einen Eingang und einen Ausgang. Zu Simulationszwecken wird alle zwei Sekunden eine Zeichenkette mit 50 Zeichen – zufällig gewählt – aus dem Bereich O (OK), A (Alkohol), K (Koffein), T (Teein) und N (Nikotin) generiert. Der Eingang empfängt die originäre Zeichenkette mit 50 Zeichen. Der Ausgang gibt die gefilterte Zeichenkette (mit nur O) aus. Es existieren drei Filter (i) F1: nur A, (ii) F2: K oder T und (iii) F3: nur N. Jedem Filter ist ein Container (i) F1 > C1, (ii) F2 > C2 und (iii) F3 > C3 für die Sammlung mit einer maximalen Kapazität von 100 Zeichen zugeordnet. Das Nervensystem beobachtet kontinuierlich den Füllgrad jeden Containers. Für jeden Stoff A, K, T und N existiert ein eigenes Verfahren für die Neutralisierung. Sobald ein Container einen Füllgrad von 90 % erreicht hat, aktiviert das Nervensystem das Verfahren. Das ausgewählte Verfahren besucht die entsprechenden Elemente und löscht (neutralisiert) diese in dem Container.

T05 // Notausgang

Ein Notausgang in einem Gebäude verfügt über zwei Sperren. Eine Sperre kann den Status locked oder unlocked haben. Standardmäßig haben beide Sperren den Status locked. Der Notausgang beobachtet kontinuierlich den Status der Brandmeldezentrale (BMZ). Sobald die BMZ einen Feueralarm meldet, wird die erste Sperre vom Notausgang automatisch auf den Status unlocked gesetzt. Neben dem Notausgang existiert ein Button. Der Button kann den Status deactivated, activated und pressed annehmen. Standardmäßig ist dieser Button deaktiviert. Auch dieser Button beobachtet kontinuierlich die BMZ. Bei einem Feueralarm wird dieser Button auf den Status activated gesetzt. Nur ein aktivierter Button kann gedrückt werden. Ein Kontrollcenter beobachtet kontinuierlich den Button. Sobald der Button gedrückt wurde, sendet das Kontrollcenter ein Kommando unlock an die Tür und die zweite Sperre wird vom Status locked auf den Status unlocked gesetzt. Nachdem beide Sperren den Status unlocked besitzen, öffnet sich der Notausgang nach zwei Sekunden automatisch. Der Notausgang existiert in den Varianten N1 und N2.

T06 // Blut-Hirn-Schranke

Eine Blutsequenz ist als zufällig gewählte Zeichenkette mit 5 Zeichen aus dem Bereich A, B kodiert. Zwischen dem Rückenmark und dem Gehirn wird die transferierte Blutsequenz durch einen zentralen Proxy überwacht. Auf dem Proxy sind die gültigen Sequenzen AABBB, BBAAB und BAAAA registriert. Für jede gültige Sequenz steht ein eigener Filter zur Verfügung. Ungültige Sequenzen gelangen nicht zum Gehirn bzw. werden herausgefiltert und unter Quarantäne gestellt. Die Quarantäne ist mit dem Gehirn verbunden. Bei einer ungültigen Sequenz sendet die Quarantäne ein Kommando ALARM an das Gehirn. Das Gehirn sendet eine T-Helferzelle zwecks Vernichtung der ungültigen Sequenz in der Quarantäne. Die T-Helferzelle übernimmt die ungültige Sequenz und veranlasst die Löschung aus der Quarantäne. Mit einem Verfahren wandelt die T-Helferzelle die Zeichenkette in ***** um.

Ein Kraftwerk ist charakterisiert durch id vom Typ int und hat vier Blöcke. Eine Turbine ist charakterisiert durch id vom Typ int, name vom Typ String und power vom Typ int. Eine Turbine kann gestartet und gestoppt werden. In jedem Block befinden sich vier Turbinen. Eine Turbine existiert in den Varianten T1 und T2. Die in einem Kraftwerk genutzte Variante ist homogen (beispielsweise nur T1). In dem Kraftwerk sind zwei Administratoren A1 und A2 permanent anwesend. Ein Administrator ist charakterisiert durch id vom Typ int, name vom Typ String und password vom Typ String. In einer Konfigurationsdatei sind die Administratoren mit den korrespondierenden Passwörtern definiert. Die Passwörter in der Konfigurationsdatei sind mit RSA oder AES verschlüsselt. In der Konfigurationsdatei ist vermerkt, mit welchem Algorithmus das korrespondierende Passwort verschlüsselt wurde. Der Zugriff von einem Administrator auf die Turbinen wird über einen Proxy kontrolliert. Administrator A1 hat Zugriff auf Block 1 und 2, A2 auf Block 3 und 4. Der Administrator meldet sich hierzu mit seinem Passwort an dem entsprechenden Proxy an. Der Proxy entschlüsselt das Passwort zu dem Administrator aus der Konfigurationsdatei und vergleicht dies mit dem übermittelten Passwort. Ist das Passwort korrekt, kann der Administrator von der Konsole über den Proxy die Kommandos start <id> oder stop <id> absetzen. Ist das Passwort nicht korrekt, erscheint die Fehlermeldung „permission denied“.

Ein Tresor – für die Aufbewahrung von wertvollen Java-Büchern – verfügt über eine Konsole und ein Lesegerät für einen elektronischen Schlüssel. Standardmäßig ist die Konsole deaktiviert. Der elektronische Schlüssel wird als Matrix mit 50 Zeilen und 20 Spalten dargestellt. Insgesamt hat die Matrix somit 1000 Elemente. Ein Element kann den Wert 0 oder 1 annehmen. Die Elemente sind von oben links nach unten rechts aufsteigend von 1 bis 1000 indiziert. Wenn der jeweilige Index ein mindestens zweistelliges Primzahl-Palindrom ist, wird der Wert 1 gesetzt, ansonsten der Wert 0. Das erste Element mit dem Wert 1 hat somit den Index 11. Auf der Konsole ist der korrekte elektronische Schlüssel zum Vergleich gespeichert. Die Konsole beobachtet kontinuierlich das Lesegerät für den elektronischen Schlüssel. Ist der elektronische Schlüssel korrekt, wird die Konsole aktiviert. Zwischen der Konsole und den beiden Stahltüren ist aus Sicherheitsgründen ein Proxy geschaltet. Nur bei aktivierter Konsole ist die Eingabe des 10-stelligen Code zum Öffnen der Türen möglich. Der 10-stellige Code lautet I8JK77NH6K und ist auf dem Proxy gespeichert. Die Konsole sendet die Eingabe des Codes zwecks Prüfung an den Proxy. Ist der eingegebene Code korrekt, sendet der Proxy das Kommando OPEN an die beiden Stahltüren. Die beiden Stahltüren öffnen sich. Wird der eingegebene Code dreimal hintereinander nicht korrekt eingegeben, sendet der Proxy das Kommando DEACTIVATE an die Konsole und das Kommando LOCK an die beiden Stahltüren. Die Konsole wird deaktiviert und die beiden Stahltüren werden verriegelt.

Ein Hochregallager ist in zwei Seiten (getrennt durch einen Gang) aufgeteilt. Auf jeder Seite existieren fünf Fächer. Jedes Fach besteht aus vier Unterfächern (vorne links, vorne rechts, hinten links und hinten rechts). Gegenstände werden nach drei Typen (normal, dangerous und fragile) unterschieden. Ein Unterfach nimmt einen bestimmten Typ eines Gegenstands auf. Die beiden hinteren Fächer können je einen Gegenstand vom Typ N aufnehmen, das Fach vorne links ein Gegenstand vom Typ D und das Fach vorne rechts ein Gegenstand vom Typ F. Für das Hochregallager existieren drei Iteratoren (i) pro Seite wird der Inhalt von jedem Fach mit den Unterfächern angezeigt, (ii) Gegenstände vom Typ N, (iii) Gegenstände vom Typ D oder F. Jedes Fach wird von einem Kontrollcenter kontinuierlich bezüglich Veränderungen (Eingang oder Ausgang) beobachtet. Zu Simulationszwecken wird das Hochregallager zu 80% (65% sind Gegenstände vom Typ N, 10% sind Gegenstände vom Typ F und 5% Gegenstände vom Typ D) befüllt. Jede Sekunde verlassen 2-3 zufällig gewählte Gegenstände das Lager und 1-3 Gegenstände mit zufällig gewählten Typ kommen hinzu. Sofern für den neuen einzulagernden Gegenstand kein geeignetes Fach zur Verfügung steht, wird dieser in eine Warteschlange eingereiht. Das Kontrollcenter verwaltet diese Warteschlange. Sofern ein geeignetes Fach frei wird, wird der Gegenstand aus der Warteschlange entfernt und in das korrespondierende Fach eingelagert.

Zu Simulationszwecken wird eine Person als eine Matrix mit 1000 x 1000 zufällig gewählten Zeichen aus dem Bereich [a-z] dargestellt. Bei den Zeichen für nicht erlaubte Gegenstände wird unterschieden zwischen k (knife), w (weapon) und d (drugs). Nachfolgend sind die Wahrscheinlichkeiten für die Zeichen k, w und d aufgeführt (i) k: 0,0005, (ii) w: 0,0001 und (iii) d: 0,000025. Ein Körperscanner am Flughafen (wie beispielsweise ProVision) verfügt über zwei Türen (Schleuse), die bei einem Mediator zentral registriert sind. Die vordere Tür ist geöffnet und die hintere Tür geschlossen. Eine Person betritt den Scanner, die vordere Tür schließt sich und der Scan beginnt. Wird bei der Person keines der Zeichen k, w oder d gefunden, öffnet sich die hintere Tür und die Person verlässt den Körperscanner. Nachdem die Person den Körperscanner verlassen hat, schließt sich die hintere Tür und die vordere Tür öffnet sich. Eine neue Person betritt den Körperscanner und der Scan startet. Ein Kontrollcenter bei der Bundespolizei beobachtet kontinuierlich den Status des Scans. Sobald bei einer Person eines der drei Zeichen k, w oder d gefunden wurde, bleibt die vordere und hintere Tür geschlossen und ein Polizist bewegt sich zu dem Scanner. Nachdem der Polizist bei dem Scanner eingetroffen ist, wird die vordere Tür geöffnet und der Mensch festgenommen.

Ein Repository besteht aus zwei Bereichen (i) temporary und (ii) primeNumbers. Der zweite Bereich besteht aus vier Fächern (i) palindrome, (ii) permutable, (iii) left-truncable und (iv) right-truncable. Zu Simulationszwecken werden alle 30 Sekunden 100 Zufallszahlen aus dem Bereich 1 bis 100000 generiert und im ersten Bereich gespeichert. Der zweite Bereich beobachtet den ersten Bereich. Sobald die neuen Zahlen im ersten Bereich eingefügt wurden, werden im nächsten Schritt die Primzahlen ermittelt und in den zweiten Bereich in die korrespondierenden Fächer verschoben. Duplikate in einem Fach im zweiten Bereich sind nicht zugelassen. Die verbleibenden Nicht-Primzahlen im ersten Bereich werden gelöscht. Für die Prüfung, ob es sich um eine Primzahl handelt, stehen die Strategien (i) Native und (ii) Miller Rabin zur Verfügung. Sofern es sich um eine Primzahl handelt, wird diese aus dem ersten Bereich entfernt und im zweiten Bereich eingefügt. Ein Wissenschaftler hat über einen Proxy lesenden Zugriff auf den zweiten Bereich und kann die Werte in eine CSV-Datei exportieren.

Ein Entertainment-System hat 10 Kanäle, eine Abspieleinheit und wird über eine Fernbedienung gesteuert. Die Abspieleinheit existiert in zwei Varianten (i) V1 und (ii) V2. Die Fernbedienung unterstützt die Kommandos (i) CHANGE, (ii) START, (iii) FORWARD, (iv) BACKWARD und (v) STOP. Mit dem Kommando CHANGE wird die aktuelle Variante der Abspieleinheit durch die andere ausgetauscht. Standardmäßig ist der erste Kanal selektiert. Mit dem Kommando START wird der aktuell selektierte Kanal gestartet. Mit dem Kommando FORWARD wird der nächste Kanal selektiert und automatisch gestartet. Mit dem Kommando BACKWARD wird der vorherige Kanal selektiert und automatisch gestartet. Mit dem Kommando STOP wird das Abspielen des aktuell selektierten Kanals gestoppt.

Ein Getränkeautomat bietet Wasser, Apfelschorle und Kaffee an. Wasser kostet 20, Apfelschorle 40 und Kaffee 60. Als Geldstücke stehen 10 und 20 zur Verfügung. Der Getränkeautomat wird durch einen Mediator gesteuert. Der Kunde drückt zunächst eine der drei Tasten für Wasser, Apfelschorle oder Kaffee. Der zu zahlende Betrag wird angezeigt und der Kunde wirft die Geldstücke sukzessive mit 1-2 Sekunden Abstand ein. Der bereits gezahlte Betrag wird vom Mediator beobachtet. Ist der zu zahlende Betrag erreicht oder überschritten, wird der Geldeinwurf gesperrt und das gewählte Getränk und ggf. Restgeld ausgegeben. Aus Sicherheitsaspekten kommuniziert der Geldeinwurf mit dem Mediator über einen Proxy.

Die Luft in einem Raum wird mit einer 3-dimensionalen Matrix (Würfel) simuliert. Die Matrix wird über drei Achsen (x,y,z) erstellt. Jede Achse hat eine Länge von 100, folglich hat diese Matrix 1 Mio. Gitterpunkte. Ein Gitterpunkt kann ein Zeichen N (normal) oder S (smoke) aufnehmen. Zu Simulationszwecken wird die 3-dimensionalen Matrix alle 5 Sekunden mit neuen Werten befüllt. Mit einer Wahrscheinlichkeit von 0,005 hat ein Gitterpunkt den Wert S. Nach der Befüllung mit neuen Werten analysiert ein Rauchmelder die Matrix. Standardmäßig ist beim Rauchmelder das Attribut *isAlarm* auf den Wert false gesetzt. In folgenden Fällen wird der Wert bei dem Attribut *isAlarm* auf true gesetzt: (i) drei S in mindestens einer Reihe hintereinander auf einer Achse, (ii) zwei S mit einem Abstand von einer weniger als drei (beispielsweise ...NNSNNSN...) in mindestens einer Reihe auf einer Achse oder (iii) fünf S in mindestens einer Reihe auf einer Achse.

T15 // Entertainment System II

In einem Entertainment-System (ES) sind zu Simulationszwecken zwei Bereiche mit je 20 Musiktiteln und Filme gespeichert. Das ES wird über eine Fernbedienung gesteuert. Die Fernbedienung hat vier Kommandos START, FORWARD, BACKWARD und STOP. Standardmäßig ist der Bereich Musiktitel ausgewählt und der Status 0 gesetzt. Die Kommandos FORWARD und BACKWARD im Status 0 ermöglichen den Wechsel zwischen den Bereichen (i) Musiktitel und (ii) Filme. Bei dem Kommando START wechselt das ES in den Status 1. Das Kommando FORWARD im Status 1 wechselt zum nächsten Eintrag. Das Kommando BACKWARD im Status 1 wechselt zum vorherigen Eintrag. Für Musiktitel und Filme existieren jeweils separate Iteratoren. Wird beim letzten Eintrag das Kommando FORWARD selektiert, wird der erste Eintrag in der Liste selektiert. Wird beim ersten Eintrag das Kommando BACKWARD selektiert, wird der letzte Eintrag in der Liste selektiert. Bei dem Kommando STOP wechselt das ES in den Status 0.

T16 // Abfüllmaschine für Mais

In einem zentralen Silo befinden sich 100 kg Maiskörner. Ein Maiskorn wiegt – zufällig bestimmt – zwischen 1-3 Gramm. Zu Simulationszwecken wird die Oberfläche von einem Maiskorn als Matrix mit 100x100 Zeichen dargestellt. Ein Zeichen ist O (OK) oder N (Nicht OK). Standardmäßig ist ein Zeichen O. Mit einer Wahrscheinlichkeit von 0,005 ist ein Zeichen N. Ein Maiskorn hat den Status O (OK) und D (DAMAGED). In den Fällen (i) vier Zeichen N waagerecht nebeneinander, (ii) vier Zeichen N senkrecht untereinander oder (iii) fünf Zeichen diagonal untereinander hat ein Maiskorn den Status D. Eine Abfüllmaschine ist mit vier Slots an das Silo angeschlossen. Vor jedem Slot ist ein Filter platziert, der die Maiskörner mit dem Status D herausfiltert. Bei jedem Durchlauf werden – zufällig bestimmt – zwischen 50 und 65 Maiskörner mit dem Status O in jeden Slot eingefüllt. Die vier Slots werden zentral durch einen Mediator verwaltet. Bei einem Durchlauf wird eine Dose mit mindestens 200 Gramm Maiskörnern befüllt. Durch die gezielte Öffnung von Slots minimiert der Mediator das überschüssige Gewicht. Nach der Leerung eines Slots wird dieser wieder automatisch neu befüllt und der Prozess beginnt erneut.

Ein Ticketautomat verfügt über zwei Slots. Standardmäßig sind die beiden Slots deaktiviert. Ein Slot akzeptiert Münzen (1, 2 und 5) und der andere Slot akzeptiert Scheine (10, 20 und 50). Ein Schein wird dargestellt als Matrix der Länge 50 und Breite 20 aus dem Zeichenpool [0,9]. In den Ecken des Scheins wird der Wert wie beispielsweise 10 dargestellt. In der Mitte des Scheins befindet sich ein Sicherheitsstreifen mit einer 20-stelligen Nicht-Primzahl. Die Primzahl-Faktoren sind – eingerahmt durch den Wert 0 – auf dem Schein zufällig verteilt. Der Automat verkauft drei Typen von Tickets (i) T1 zu einem Preis von 15, (ii) T2 zu einem Preis von 24 und (iii) T3 zu einem Preis von 35. Über das Kommando `SELECT <ticketType>` wird zu Beginn ein Ticket selektiert und die beiden Slots werden aktiviert. Zwischen den Slots und der Geldkassette ist aus Sicherheitsgründen ein Proxy geschaltet. Der Proxy prüft die Münzen und Scheine auf Gültigkeit. Eine Münze 3 ist beispielsweise ungültig und wird zurückgegeben. Im Abstand von 3-5 Sekunden werden Münzen oder Scheine den Slots zugeführt. Auf dem Proxy wird die Summe der gültigen Münzen/Scheine vermerkt. Die beiden Slots beobachten kontinuierlich den Status auf dem Proxy. Sobald der zu zahlende Betrag erreicht ist, werden die beiden Slots vom Proxy deaktiviert, ein Beleg (Datum/Uhrzeit, ausgewähltes Ticket und Preis, gezahlter Betrag und Restgeld) erstellt.

Ein Ticketautomat verfügt über ein Touchpad, einen Coin-Slot und eine Bezahlkomponente. Das Touchpad hat drei Buttons (i) ADD, (ii) OK und (iii) CANCEL. Das Drücken eines Buttons auf dem Touchpad generiert ein Kommando an die Bezahlkomponente. 15 Minuten Parken kosten 3 Euro. Das Attribut `value` bei der Bezahlkomponente ist standardmäßig 0. Durch das Kommando ADD wird die Parkdauer um 15 Minuten bzw. das Attribut `value` um 3 Euro erhöht. Wird der Button ADD beispielsweise zweimal gedrückt, ist die Parkdauer 30 Minuten und `value` 6 Euro. Standardmäßig ist der Coin-Slot deaktiviert. Wird der Button OK gedrückt, ist der Button ADD deaktiviert und der Coin-Slot wird von der Bezahlkomponente durch den Befehl PAY aktiviert. Sobald der Coin-Slot aktiviert wurde, beobachtet die Bezahlkomponente kontinuierlich den Münzeinwurf. Ein Kunde wirft im Abstand von 1-2 Sekunden (zufällig bestimmt) verschiedene Münzen (1 oder 2) in den Coin-Slot. Sobald der zu zahlende Betrag erreicht oder überschritten wurde, wird der Coin-Slot von der Bezahlkomponente durch den Befehl PAYED deaktiviert. Danach erstellt die Bezahlkomponente eine Quittung mit Datum, Uhrzeit, Parkdauer, Parkende, Betrag, gezahltes Geld und ggf. Rückgeld.

Ein Kunde ist charakterisiert durch id vom Typ id und name vom Typ String. In einer speicherbasierten Datenbank werden die Kunden gespeichert. Bei der Datenbank gibt es zwei Rollen R1 und R2. Benutzer U1 hat die Rolle R1. Benutzer U2 hat die Rolle R2. Die Benutzer sind zentral auf einem Proxy registriert. Ein Benutzer meldet sich mit benutzername/passwort auf dem zentralen Proxy an. Nach erfolgreicher Authentifizierung kann der Benutzer Kommandos absetzen. R1 darf die Kommandos INSERT, UPDATE, DELETE und UNDO ausführen. R2 darf das Kommando SELECT ausführen. Das Kommando INSERT fügt einen neuen Datensatz ein: INSERT (1,Mueller) Sofern ein Kunde mit der gleichen id bereits existiert, wird INSERT mit einer Fehlermeldung abgewiesen. Mit dem Kommando UPDATE wird ein bestehender Datensatz aktualisiert: UPDATE (1,Maier) – name Mueller wird geändert nach Maier. Sofern kein Datensatz mit der id existiert, wird das UPDATE mit einer Fehlermeldung abgewiesen. Mit den Kommandos DELETE (1) oder DELETE (1,Maier) wird ein Datensatz gelöscht. Sofern kein Datensatz mit der id existiert, wird das DELETE abgewiesen. Mit UNDO können die abgesetzten Kommandos sukzessive widerrufen werden. Das Kommando SELECT zeigt alle Datensätze aufsteigend sortiert nach der id an.

T20 // Gehörschutz

Ein Gehörschutz hat einen Schalter für den Statuswechsel (S0, S1, S2 und S3) und ist mit zwei EarSets ausgestattet. Ein Ton wird als Zeichenkette mit 100 zufälligen Zeichen aus dem Pool [A, B, H, T] dargestellt. Vier verschiedene Filter stehen zur Verfügung (i) jedes A, (ii) maximal fünf H werden weitergeleitet, (iii) 2 T unmittelbar nacheinander und (iv) jedes zweite B. Es werden zwei EarSets eingesetzt. EarSetA besitzt die Filter (i) und (ii). EarSetB besitzt die Filter (iii) und (iv). Der Ausgangsport von EarSetA hat zwei Pin. Der Port für den Eingang von EarSetB hat drei Pin. EarSetA und EarSetB können mit einem Adapter miteinander kombiniert werden. In diesem Fall trifft der Ton auf EarSetA und der gefilterte Ton wird durch die Ports bzw. den Adapter zu EarSetB weitergeleitet. Initial befindet sich der Gehörschutz im Status S0 und der Ton trifft ungefiltert auf das Ohr. Schalter betätigt → S1: EarSetA; Schalter betätigt → S2: EarSetB; Schalter betätigt → S3: EarSetA und EarSetB; Schalter betätigt → S0: keine Filter.

T21 // Automat für Pizza

Am Flughafen ist ein innovativer Automat für Pizza aufgestellt. Es stehen vier Tasten (i) select, (ii) next, (iii) undo und (iv) pay zur Verfügung. Die Kosten sind in Klammern angegeben. Die Basis ist Käse (5). Zusätzlich können die Beläge Salami (1), Paprika (1) und Parmaschinken (2) hinzugefügt werden. Mit select wird ein Belag ausgewählt. Mit next wird in der Liste der Beläge navigiert. Mit undo wird der aktuell hinzugefügte Belag verworfen und vorherige Auswahl wiederhergestellt. Mit pay wird bezahlt und die Pizza mit ggf. selektierten Belägen gebacken. Nach drei Sekunden ist die Pizza fertig und wird bereitgestellt. Der Kunde entnimmt die Pizza und es wird eine Quittung mit den Einzelpositionen (Basis + ggf. Zutaten), dem Gesamtbetrag sowie Datum/Uhrzeit erstellt.

Ein elektronischer Atemschutz hat einen Schalter und zwei Kanäle. In jedem Kanal findet sich ein Filter für die einströmende Luft. Zwecks Simulation der einströmenden Luft wird alle fünf Sekunden eine Zeichenkette mit 50 zufällig gewählten Zeichen aus dem Bereich A, B und C generiert. Bei zwei geöffneten Kanälen wird die Zeichenkette in zwei Hälften mit je 25 Zeichen für den linken und rechten Kanal aufgeteilt. Standardmäßig befindet sich der Atemschutz im Status 0 und beide Kanäle sind geschlossen. Wird der Schalter einmal gedrückt, befindet sich der Atemschutz im Status 1 und der linke Kanal ist geöffnet. Wird der Schalter erneut gedrückt, befindet sich der Atemschutz in Status 2 und der linke und rechte Kanal sind geöffnet. Wird der Schalter erneut gedrückt, befindet sich der Atemschutz im Status 0 und beide Kanäle sind geschlossen. Der Filter existiert in zwei dynamisch austauschbaren Varianten (i) F1 und (ii) F2. F1 entfernt alle A und B. F2 entfernt alle B und C. Die verwandten Filter für den Atemschutz sind homogen (für beide Kanäle gleich) und dynamisch austauschbar.

Ein Stausee zur Energie-Erzeugung hat einen maximalen Füllstand von 1000 Einheiten. Der initiale Füllstand ist 0. Zu Simulationszwecken fließen – zufällig bestimmt – jede Sekunde 80-120 Einheiten zu und 60-100 Einheiten ab. Der Staudamm hat zwei Tore (i) G1 u. G2. Jedes Tor beobachtet je mit einem Sensor den aktuellen Füllstand des Stausee. Ist ein Füllstand von 90% erreicht, öffnet sich G1 und 50 Einheiten fließen sofort ab. Ist ein Füllstand von 95% erreicht, öffnet sich G2 und 100 Einheiten fließen sofort ab. Zusätzlich kann der Staudamm von einer Konsole im Kontrollcenter bedient werden. Der Administrator meldet sich mit seinem Namen und Passwort auf dem Proxy an. Die Administratoren A1 und A2 sind berechtigt, das Tor G1 zu öffnen. Hierzu sendet der Administrator A1 oder A2 über die Konsole das Kommando OPEN an das Tor G1. Beim Öffnen des Tors G1 fließen sofort 50 Einheiten ab. Der Administrator A3 ist berechtigt beide Tore G1 und G2 gleichzeitig zu öffnen. Beim Öffnen der beiden Tore G1 und G2 fließen sofort 100 Einheiten ab. Hierzu sendet der Administrator A3 das Kommando OPEN an die Tore G1 und G2. Nach einer Sekunde sendet die Konsole automatisch das Kommando CLOSE an Tor G1 und ggf. Tor G2.

Zu Beginn und am Ende der Landebahn sind auf der linken und rechten Seite je ein Messgerät für Wind aufgestellt. Alle vier Messgeräte werden zentral von einer Messwarte kontinuierlich beobachtet. Ein – dynamisch austauschbares – Messgerät existiert in der Variante M1 und M2. Der Wind wird mit einem Zeichen aus dem Satz [<,>] dargestellt. Die Zeichenkette für die Simulation des Windes wird alle 3 Sekunden zufällig neu generiert. Die Messwarte ist mit dem Landebahnmanagement verbunden. Eine Warnung wird als Kommando von der Messwarte an das Landebahnmanagement in folgenden zwei Fällen kommuniziert: (i) vorne links > - hinten rechts < (ii) vorne rechts <, hinten links >. Es stehen die Kommandos clear und warning zur Verfügung.

Die Mikrowelleneinheit existiert in zwei Varianten W1 und W2. W1 hat eine Leistungsfähigkeit von 2 und W2 eine Leistungsfähigkeit von 3. Der Garraum wird dargestellt als 3-dimensionale Matrix mit der Länge 100, Höhe 50 und Breite 30. Standardmäßig ist der Garraum mit dem Zeichen 0 gefüllt. Die Mikrowelle wird über eine Fernbedienung mit Kommandos gesteuert. Das Kommando OPEN öffnet die Tür. Mit dem Kommando PUT(objekt) wird das zu erhitzende Objekt mittig in den Garraum gelegt. Mit dem Kommando CLOSE wird die Tür geschlossen. Mit dem Kommando START wird die Erhitzung gestartet. Das zu erhitzende Objekt wird als 3-dimensionale Matrix mit der Länge 20, Höhe 25 und Breite 10 dargestellt. Das Objekt ist mit dem Zeichen 2 gefüllt. Die Erhitzung dauert 6 Sekunden und jede Sekunde wird der aktuelle Wert von dem Objekt mit dem Wert der Mikrowelle (2 oder 3) multipliziert. Bei der Leistungsfähigkeit 2 wird beispielsweise das Objekt jede Sekunde wie folgt erhitzt: 4, 8, 16, 32, 64 und 128. Der Stromanschluss der Mikrowelle hat 3 Pole. Die Steckdose für Strom hat 2 Pole. Für den Anschluss der Mikrowelle an die Steckdose wird eine Adapter verwandt.

Ein Weizenfeld wird dargestellt als Matrix mit Länge 1000 und Breite 100. Eine Weizenpflanze W wird dargestellt als Matrix mit Länge 40-45 (zufällig bestimmt) und Breite 5. W besteht aus zwei Abschnitten (unten und oben). Der obere 2. Abschnitt hat eine feste Länge von 30 und wird dargestellt mit Zeichen aus dem Pool [c,x]. Der untere 1. Abschnitt hat eine variable Länge in Abhängigkeit der Länge des 2. Abschnitts und wird dargestellt mit Zeichen aus dem Pool [a,b,d,e]. Der Mähdrescher bearbeitet pro Sekunde einen Bereich der Länge 100 und Breite 25 (2500 W). Blöcke zu 50 W werden gedroschen und die Zeichen c fallen in den Behälter I und die anderen Zeichen in den Behälter II. Behälter I hat ein Volumen von 25000 Zeichen. Behälter II hat ein Volumen von 50000 Zeichen. Eine Kontrolleinheit im Mähdrescher beobachtet kontinuierlich die beiden Behälter I und II. Ist bei Behälter I das maximale Volumen erreicht, wird der gesamte Inhalt in einen Anhänger geladen und die Bearbeitung fortgesetzt. Ist bei Behälter II das maximale Volumen erreicht, wird zunächst der Inhalt komprimiert (mindestens drei gleiche Zeichen; Beispiel: aaa → 3a*). Ein Strohballen wird dargestellt als Würfel der Länge 10. Es werden vollständige Strohballen erstellt, ggf. ein Rest verbleibt im Behälter II. Der Mähdrescher wird über die Kommandos MOVE, TURN und STOP von einem Kontrollcenter gesteuert.

Es existieren drei Hauptverzeichnisse R1, R2 und R3. Ein Hauptverzeichnis hat drei Unterverzeichnisse Sx. In einem Unterverzeichnis existieren 2-5 „Dateien“ mit 1000 zufälligen Zeichen aus dem Pool [a-z,0-9]. Die Signaturen, lcky, wcry und ptya sind bekannt. Pro Sekunde werden bei einer zufällig selektierten Datei 250 beliebige Zeichen vertauscht und mit einer Wahrscheinl. von 0.01 eine zufällig gewählte Signatur an einer beliebigen Stelle platziert. Für die Suche stehen die Algorithmen Boyer-Moore, Knuth-Morris-Pratt und Rabin-Karp zur Verfügung. Der Virens Scanner beobachtet kontinuierlich alle „Dateien“ bezüglich Veränderung und scannt automatisch die Datei.

Ein Flugzeug ist mit vier Triebwerken vom Typ RR Trent 900 ausgestattet. Das Triebwerk existiert in zwei Varianten „RR Trent 900 N“ und „RR Trent 900 X“. Der zulässige Temperaturbereich bei der Variante „RR Trent 900 N“ ist zwischen 0 Grad Celsius und 650 Grad Celsius. Der zulässige Temperaturbereich bei der Variante „RR Trent 900 X“ ist zwischen 0 Grad Celsius und 700 Grad Celsius. Jedes Triebwerk ist mit einem Temperatursensor ausgestattet. Zu Simulationszwecken wird jede Sekunde die Temperatur um 50 Grad Celsius synchron bei jedem Triebwerk erhöht. Jede Änderung wird automatisch textuell angezeigt. Sobald die maximal zulässige Temperatur überschritten ist, wird eine Warnmeldung angezeigt. Durch einen Adapter ist auch die Anzeige in Fahrenheit – anstelle von Grad Celsius – möglich.

T29 // Gepäckverladung

Ein Mediator steuert die Verteilung der Gepäckstücke. Ein Gepäckstück ist charakterisiert durch id, type und weight. Es stehen drei Typen von Gepäckwagen zur Verfügung (i) normal, (ii) bulky, (iii) dangerous. Ein Gepäckwagen ist durch id und type charakterisiert. Maximales Gewicht für Gepäckwagen (i) 250, (ii) 450 und (iii) 150. Gepäck wird im Bezug auf das Ziel unterschieden nach EU und Non-EU. Der Mediator verwaltet insgesamt sechs Schächte – normal-eu, normal-non-eu, bulky-eu, bulky-non-eu, dangerous-eu und dangerous-non-eu. Zu Simulationszwecken werden 250 Gepäckstücke mit zufällig gewähltem type und weight (Bereich: 25-55) generiert. Der Mediator verteilt die Gepäckstücke durch Nutzung der (kombinierten) Filter über die Schächte auf die bereitstehenden Gepäckwagen. Der Mediator beobachtet kontinuierlich das aktuelle Gewicht der Gepäckwagen. Sobald ein Gepäckwagen das nächste Gepäckstück nicht mehr aufgrund des maximal zulässigen Gewichts aufnehmen kann, wird ein neuer leerer Gepäckwagen an dem korrespondierenden Schacht bereitgestellt.

T30 // Dynamische Firewall

Eine Firewall beobachtet kontinuierlich den status eines übergeordneten Kontrollcenters. Beim status wird zwischen normal, warning und critical unterschieden. Zu Simulationszwecken scannt die Firewall eine 30-stellige Zeichenkette. Ein Zeichen kodiert einen Informationstyp. Informationstypen sind (i) E (E-Mail), (ii) V (Virus), (iii) S (Spam), (iv) A (Attachment) und (v) C (Certificate). Die Zeichenkette wird zufällig aus den Zeichen E, V, S, A und C komponiert. Für jeden Informationstyp steht ein separater Filter zur Verfügung. In Abhängigkeit von dem status werden drei Filtergruppen gebaut und dynamisch angewandt. Filtergruppen (i) normal: E (ja), V (nein), S (nein), A (ja) C (ja); (ii) warning: E (nein), V (nein), S (nein), A (ja), C (ja); critical: (iii) E (nein), V (nein), S (nein), A (nein), C (ja).

T31 // MarshallerCommand, Proxy, Mediator

Ein Marshaller lotst ein Flugzeug durch Handzeichen in die korrekte Abstellposition. Die Handzeichen move, left, right oder stop werden als Kommando dem Piloten übermittelt. Das Flugzeug wird durch einen Joystick gesteuert (move = Joystick nach vorne, stop = Joystick nach hinten). Der Pilot wendet die Befehle vom Marshaller unmittelbar auf den Joystick und damit auf die Steuerung des Flugzeuges an. Der Joystick sendet die Befehle über einen Proxy an einen Mediator, bei dem die Einheiten 4 Triebwerke, 5 Fahrwerke und 18 Bremsen registriert sind.

T32 // BoxComposite, Iterator, Filter

In einer Box werden Schrauben aufbewahrt. Es werden zwei Typen von Schrauben S1 und S2 unterschieden. Die Box besteht aus zwei großen Fächern FI und FII. FI besteht aus zwei Fächern FI1 und FI2. In FI1 und FI2 befinden sich Schrauben des Typs S1. FII besteht aus zwei Fächern FII1 und FII2. In FII1 befinden sich Schrauben vom Typ S1 und im Fach FII2 Schrauben des Typs S2. Eine Schraube hat den Zustand O (OK) oder D (damaged). In den Fächern befinden sich je 250 Schrauben. Standardmäßig ist der Zustand einer Schraube O. Mit einer Wahrscheinlichkeit von 0.005 hat eine Schraube den Status D. Mit einem Iterator und Filter werden die ggf. beschädigten Schrauben ermittelt.

T33 // KundenabfrageStrategy, Proxy, Command

Ein Kunde ist charakterisiert durch id und name. In einem zentralen Repository sind Kunden gespeichert. Für das Attribut id ist ein Index angelegt. Für die aufsteigende Sortierung der id stehen die Algorithmen (i) MergeSort und (ii) QuickSort zur Verfügung. Der Index beobachtet das Repository. Bei einer Änderung wird der Index automatisch aktualisiert. Der Benutzer ist mit username und password auf einem Proxy registriert. Die Kommunikation zwischen dem Repository und dem Benutzer erfolgt über den Proxy. Nach erfolgreicher Authentifizierung stehen dem Benutzer die Kommandos QUERY <id> und QUERY ALL. Bei QUERY ALL werden maximal 25 Kunden angezeigt. Bei QUERY <id> wird der korrespondierende Kunde über den Index und Binärsuche ermittelt.

T34 // Sicherheit in einem PlanetariumMediator, Filter, Observer

Die maximale Kapazität eines Planetariums ist wie folgt definiert (i) adults 55, (ii) children 25. Zu Beginn befinden sich keine Personen im Planetarium. Jede Sekunde betritt und verlässt eine zufällig gewählte Anzahl (Bereich: 5-15, maximal alle Personen) von adults und children über zwei Schleusen das Planetarium. Die erste Schleuse ist der Eingang und die zweite Schleuse ist der Ausgang. Eine Schleuse hat den Status open oder closed. Das Planetarium bzw. der Raum wird kontinuierlich mit einem Sensor überwacht und über zwei Filter wird die Anzahl der adults und children ermittelt. Schleusen und Sensor sind zentral bei einem Mediator registriert. Sobald die maximale Kapazität erreicht bzw. überschritten ist, wird die erste Schleuse gesperrt, die zweite Schleuse bleibt geöffnet.

Eine elektrische Zahnbürste wird über einen Schalter bedient. Durch die Betätigung des Schalter wird ein Kommando an den Bürstenkopf gesandt. Der Bürstenkopf existieren in zwei Varianten (i) H1 und (ii) H2. Initial befindet sich die Zahnbürste im Status S0. Wird im Status S0 der Schalter einmal für eine Sekunde gedrückt, befindet sich die Zahnbürste im Status S1 und der Bürstenkopf rotiert mit 300 im Uhrzeigersinn. Wird im Status S1 der Schalter einmal für eine Sekunde gedrückt, befindet sich die Zahnbürste wieder im Status S0 und die Rotation von dem Bürstenkopf stoppt. Wird im Status S1 der Schalter zweimal kurz gedrückt, befindet sich die Zahnbürste im Status S2 und der Bürstenkopf rotiert mit 300 gegen den Uhrzeigersinn. Wird im Status S2 der Schalter einmal für eine Sekunde gedrückt, befindet sich die Zahnbürste wieder im Status S0 und die Rotation von dem Bürstenkopf stoppt. Wird im Status S2 der Schalter zweimal kurz gedrückt, befindet sich die Zahnbürste im Status S1 und der Bürstenkopf rotiert mit 300 im Uhrzeigersinn.

Ein Gebäude hat 10 Etagen. Ein Raum hat den Typ O, M, C oder T. Jede Etage hat 30 Räume (25 x O, 3 x M, 1 x C und 1 x T), die durch einen Gang in zwei Seiten (links / rechts) mit je 15 Räumen aufgeteilt sind. Die Verteilung der Räume auf die Seiten ist zufällig. Für das sukzessive Durchlaufen des Räume existieren sieben Iteratoren (i) Etage, (ii) Typ O, (iii) Typ M, (iv) Typ C, (v) Typ T, (vi) linke Seite und (vii) rechte Seite. Die Räume werden von einem Roboter gereinigt. Für jeden Typ des Raums verfügt der Roboter über ein spezielles Verfahren. Über eine Fernbedienung und Kommandos wird der Roboter bedient. Das Kommando START <procedure> startet die Reinigung. Für den Parameter procedure stehen die Optionen (i) FLOOR, (ii) O, (iii) M, (iv) C, (v) T, (vi) LEFT und (vii) RIGHT zur Verfügung.

Zu Simulationszwecken wird Ultrakurzwelle (UKW) als Kette der Länge 1000 mit zufällig gewählten Zeichen aus dem Bereich [0,1] dargestellt. Die Frequenzen (i) 20, (ii) 35 und (iii) 60 entsprechen der Anzahl von 1 in der Zeichenkette. Mikrowelle wird mit Zeichen aus dem Bereich [0-9] dargestellt. Das Radio wandelt über einen Adapter UKW in Mikrowelle um. Zunächst werden Blöcke zu je 50 Zeichen gebildet und nach binär umgewandelt: 010110011011000101010011010011011011-00110110000101 → 394472566738309. Im nächsten Schritt werden die Primzahl-Faktoren gebildet: 719 6221 881917. Die Primzahl-Faktoren werden – getrennt durch 0 – zu einer Zeichenkette zusammengesetzt: 719062210881917. Für jeden Block wird eine Prüzfiffer mit dem Luhn-Algorithmus ermittelt und am Ende des Block angefügt. Die 20 Blöcke werden zusammengesetzt und vom Radio verarbeitet. Das Radio hat zwei Schalter (i) On/Off und (ii) Next. Initial befindet sich das Radio im Status S0 und ist ausgeschaltet. Bei Betätigung des ersten Schalter wird das Radio angeschaltet und die erste Frequenz 20 ausgewählt (Status S1). Wird der erste Schalter im Status S1 betätigt, wird das Radio ausgeschaltet. Wird im Status S1 der zweite Schalter betätigt, wird die nächste Frequenz ausgewählt. Bei Betätigung des zweiten Schalter bei selektierter Frequenz 60, wird die Frequenz 20 (Start) ausgewählt.

T38 // AutovermietungIterator, Command, Observer

Eine Autovermietung bietet die Fahrzeugtypen (i) V1, (ii) V2, (iii) V3, (iv) V4 und (v) V5 an. Zu jedem Fahrzeugtyp existiert ein Iterator für das sukzessive Durchlaufen und je drei Fahrzeuge in einem zentralen Repository. Mit dem Kommando RENT <typ> wird ein Fahrzeugtyp gebucht und mit RETURN <id> das Fahrzeug zurückgegeben. Sofern ein Fahrzeugtyp nicht verfügbar ist, wird eine Reservierung für den Kunden angelegt. Sobald ein gebuchtes Fahrzeug des reservierten Typs zurückgegeben wird, erfolgt automatisch die Buchung und der Kunde wird informiert.

T39 // TaucheruhrCommand, State, Memento

Eine Taucheruhr ist in zwei Bereiche (i) A1 und (ii) A2 unterteilt. Folgende Informationen (i) aktuelle Uhrzeit, (ii) aktuelle Dauer des Tauchvorgangs, (iii) aktuelle Tiefe, (iv) verbleibender Sauerstoff in den zwei Tauchflaschen und (v) Haie im Umkreis von 100 Metern werden angezeigt. A1 sind (i), (ii) und (iii) zugeordnet. A2 sind (iv) und (v) zugeordnet. Die Tauchflaschen haben je ein Fassungsvermögen von 1000 Einheiten und pro Sekunde werden 50 Einheiten je Flasche parallel verbraucht – nach 20 Sekunden ist der Inhalt beider Tauchflaschen verbraucht. Die Taucheruhr verfügt über zwei Buttons (i) B1 und (ii) B2. Durch einmaliges Drücken von B1 wird zwischen den Bereichen A1 und A2 gewechselt. Durch einmaliges Drücken von B2 wird die nächste Information innerhalb des selektierten Bereichs ausgewählt. Initial ist die erste Information in dem ausgew. Bereich selektiert.

T40 // Primzahlen IDecorator, Strategy, Builder

Für den Bereich 1 bis 1000000 werden die Primzahlen über die Methode primes ermittelt. Die Methode histogram erstellt eine textuelle Übersicht für die Verteilung der Endziffern von 1 bis 9. Die beiden Methoden primes und histogram werden mit einem Decorator zusammengesetzt: histogram(primes(data)). Mit Unterstützung eines Builder wird die textuelle Übersicht generiert.

T41 // Primzahlen IIStrategy, Proxy, Command

Für die Überprüfung von Primzahlen stehen die beiden Algorithmen SieveOfEratosthenes und MillerRabin als Strategien zur Verfügung. Das Ergebnis der Prüfung wird am Bildschirm angezeigt. Mit dem MersenneTwister werden 1000 unterschiedliche Zufallszahlen aus dem Wertebereich 100 bis 1000000 generiert. Werte ≤ 10000 werden mit dem Algorithmus SieveOfEratosthenes überprüft. Werte > 10000 werden mit dem Algorithmus MillerRabin überprüft. Mit dem Kommando GENERATE werden die Zufallszahlen generiert und in einer CSV-Textdatei aufsteigend sortiert gespeichert. Mit dem Kommando ANALYSE wird auf die bestehende CSV-Textdatei zugegriffen und die Analyse gestartet. Findet ANALYSE keine existierende CSV-Datei, wird diese vorher durch das Kommando GENERATE generiert. Auf einem Proxy sind die beiden Rollen R1, R2 sowie die Benutzer U1, U2 gespeichert. Eine Rolle ist charakterisiert durch id vom Typ Integer und name vom Typ String. Ein Benutzer ist charakterisiert durch id vom Typ Integer, name vom Typ String und role vom Typ Role. R1 darf nur das Kommando GENERATE ausführen. R2 darf die Kommandos GENERATE und ANALYSE ausführen. U1 hat die Rolle R1. U2 hat die Rolle R2. U1 oder U2 können sich mit dem Passwort 1234 auf dem Proxy anmelden und Kommandos ausführen.

Ein Pilot kann textuelle Informationen in einem intelligenten elektronischen Notizbuch festhalten. Durch die Betätigung der Tastenkombination [Strg]+[Z] kann die letzte Eingabe rückgängig gemacht werden. Das Notizbuch existiert in zwei Varianten (i) maximal drei Versionen und (ii) maximal fünf Versionen werden für die Wiederherstellung vorgehalten. Des Weiteren wird die Erfassung kontinuierlich auf zu vervollständigende Wörter beobachtet. Standardmäßig werden folgende Wörter unterstützt: rwy → runway und dp für descent point. Die Auto-Vervollständigung wird durchgeführt, sobald hinter dem (Schlüssel-)Wort ein Leerzeichen vorhanden ist. Bei dem elektronischen Notizbuch handelt es sich um eine Kommandozeilen-Applikation (keine GUI).

Ein Flugzeug wird standardmäßig mit dem Wert ABC für den Parameter type vom Typ string ausgeliefert. Der Wert ist dreistellig und wird mit Zeichen aus dem Satz [A,B,C] gebildet. Im Rahmen der Wartung kann der Wert für type von einem registrierten Techniker verändert werden. Der Techniker meldet sich an einer Konsole an. Die Veränderung wird in einer Historie zwecks Nachvollziehbarkeit und Wiederherstellung festgehalten. Von einem zentralen Kontrollcenter wird die Änderungsfrequenz kontinuierlich beobachtet. Beträgt zum dritten Mal die Differenz vom vorherigen zum aktuellen Wert mindestens zwei Zeichen, erhält der Mitarbeiter im Kontrollcenter eine Warnung, dieser meldet sich an der Konsole an und setzt über Memento den Standardwert.

Ein Fahrstuhl existiert in den Varianten E1 und E2 und hat die Tasten closeDoor, openDoor, move sowie stop. Beim Drücken einer der drei Tasten wird ein Kommando für die jeweilige Aktion an den Fahrstuhl gesandt. In Bezug auf den Status des Fahrstuhls werden doorClosed, doorOpen, inMove und stopped unterschieden. Standardmäßig befindet sich der Fahrstuhl im Status doorOpen. In diesem Status kann nur die Taste closeDoor betätigt werden. Bei Betätigung der Taste closeDoor wechselt der Fahrstuhl in den Status doorClosed. Im Status doorClosed können die Tasten move oder openDoor betätigt werden. Bei Betätigung der Taste move wechselt der Fahrstuhl in den Status inMove, danach in den Status stopped und doorOpen.

Ein Kaffeeautomat an einem Flughafen bietet die Sorten Espresso, Americano und Cappuccino an. Der Automat wird mit einem Knopf bedient. Initial ist die Maschine ausgeschaltet und im Status S0. Das Drücken des Knopfs erzeugt ein Kommando an die Kontrolleinheit des Automaten. Durch das einmalige Drücken des Knopfs wird der Automat eingeschaltet und befindet sich im Status S1. Im Status S1 wird die Sorte ausgewählt. Initial ist der erste Eintrag Espresso selektiert. Durch das Drehen nach rechts oder links kann vorwärts bzw. rückwärts die Liste der Sorten sukzessive durchlaufen werden. Wird im Status S1 der Knopf einmal gedrückt, wird die Sorte selektiert und die (Initial-)Menge kann verändert werden. Der Automat befindet sich im Status S2. Durch das Drehen nach rechts wird die Menge um eins erhöht und nach links um eins reduziert. Die Initial-/Minimalmenge bei allen Sorten beträgt 50, die Maximalmenge 240.

Eine Textdatei mit 1000 beliebigen Zeichen [a-z] im Verzeichnis temp wird durch „Application -e <strategy> filename“ verschlüsselt. Für die Verschlüsselung stehen RSA und AES zur Verfügung. Nach der Verschlüsselung wird eine Logdatei (logfile) mit Name der Textdatei und MAC-Adresse des Rechners generiert (Builder). Über einen Proxy wird der Betrag 50 als MD5-Hash und die Logdatei eingegeben „Proxy -p c0c7c76d30bd3dcaefc96f40275bdc0a <logfile>“. Nach Bezahlung wird vom Proxy eine Datei mit Informationen (infofile) zur Entschlüsselung generiert und im Verzeichnis temp gespeichert. Mit „Application -d <infofile>“ wird die Entschlüsselung durchgeführt.

Zu Simulationszwecken existieren zwei Flughäfen A und B mit einer Entfernung von 1000. Gleichzeitig startet ein Flugzeug F1 in A und ein Flugzeug F2 in B. F1 und F2 sind bei der Flugsicherung registriert und werden kontinuierlich überwacht. F1 und F2 nutzen die gleiche Route und bewegen sich nach dem Start in einer Reishöhe von initial 10000. F1 und F2 fliegen konstant mit einer Geschwindigkeit von 20 pro Sekunde. Die zulässige Reishöhe liegt im Normalfall zwischen 9500 und 11250. Der Mindestabstand zwischen F1 und F2 bezüglich der Höhe beträgt 250. Mit einer Wahrscheinlichkeit von 0.2 ändert ein Flugzeug pro Sekunde die Reishöhe um +/- 250 innerhalb des zulässigen Bereichs. Sind die beiden Flugzeuge eine Sekunde voneinander entfernt und auf gleicher Reishöhe, wird die Flugsicherung automatisch benachrichtigt. Die Flugsicherung sendet das Kommando CLIMB an F1 oder F2 (zufällig bestimmt). F1 oder F2 steigt um 250. In diesem kritischen Fall liegt die zulässige Reishöhe zwischen 9250 und 11500. Eine Kollision von F1 und F2 in der nächsten Sekunde wird dadurch ausgeschlossen.

Eine Schaufel aus Titan von einem Triebwerk besteht aus acht Hauptzellen vom Typ Zelle. Jede Hauptzelle besteht aus vier Subzellen vom Typ Zelle. Das Triebwerk besteht aus drei Stufen. Die erste Stufe hat 32, die zweite 48 und die dritte 64 Schaufeln. Zu Simulationszwecken wird ein Luftstrom als Kette mit Zeichen aus dem Pool [a,b] dargestellt. Die Zeichenkette hat zu 40% das Zeichen b. Die Zeichen sind zufällig verteilt. Von der Kontrolleinheit wird das Triebwerk mit START ENGINE gestartet. Initial hat das Triebwerk 1000 Umdrehungen und es wird ein Luftstrom mit 1000 Zeichen „angesogen“. Nach der ersten Stufe sind vier aufeinanderfolgende b durch 4b und drei aufeinanderfolgende a durch 3a komprimiert (Beispiele (i) abbabbbbab → abba4bab (ii) abaaaba → ab3aba). Nach der zweiten Stufe ist jedes aba durch x komprimiert (Beispiel: ab3aba → ab3x). Nach der dritten Stufe sind drei aufeinanderfolgende b durch 3b komprimiert. Jede Sekunde wird mit dem Kommando INCREASE die Anzahl der Umdrehungen und Luftstrom um 500 erhöht. Initial hat eine Subzelle die Temperatur 0. Pro Sekunde erhöht sich die Temperatur pro Subzelle um 1-3 (zufällig bestimmt). Die Temperatur der Hauptzelle ist der Median der Temperatur der assoziierten Subzellen. Die Temperatur der Schaufel ist der Mittelwert der Temperatur der assoziierten Hauptzellen. Die Temperatur der Stufe ist der Mittelwert der assoziierten Schaufeln. Jede Stufe hat einen Temperatursensor. Die Kontrolleinheit beobachtet kontinuierlich den Sensor. Sofern für eine Stufe die Temperatur von 19.5 überschritten wird, generiert die Kontrolleinheit ein Alarm. Die Simulation dauert 10 Sekunden.

Ein Dokument wird als 2-dimensionale Matrix mit 60 Zeilen (y: 0-59) und 80 Spalten (x: 0-79) dargestellt. Bei den Dokumenten werden die Typen normal, vertraulich und geheim unterschieden. Zu Simulationszwecken werden 10 Dokumente (5x normal, 3x vertraulich und 2x geheim) mit zufällig gewählten Zeichen aus dem Bereich [a-z,0-9] generiert. Für jeden Dokumententyp existiert im Drucker eine separate Einheit mit eigenem Repository. Die Dokumente werden sukzessive an die Kontrolleinheit kommuniziert und anschließend an die zuständige Einheit weitergeleitet. Dokumente vom Typ normal werden im Repository gespeichert und unmittelbar ausgegeben. Dokumente vom Typ vertraulich werden mit RSA verschlüsselt und im Repository gespeichert. Dokumente vom Typ geheim werden mit AES verschlüsselt und im Repository gespeichert. Bei Dokumenten vom Typ vertraulich und geheim gibt der registrierte Benutzer über ein Touchpad seinen Benutzernamen, das Passwort und die ID des zu druckenden Dokumentes ein (Proxy). Die Betätigung des Button Print generiert ein Kommando an den Proxy. Der Proxy leitet diese Anfrage an die zuständige Einheit weiter und nach Validierung der Eingabedaten wird das Dokument entschlüsselt und unmittelbar am Bildschirm ausgegeben.

Es werden zwei Kundentypen S und L unterschieden. Kunden vom Typ S fahren Kurzstrecke und Kunden vom Typ L Langstrecke. Ein Taxistand hat zwei Spuren. Spur I ist für den Kundentyp S. Spur II ist für den Kundentyp L. In jeder Spur stehen gleichzeitig vier Taxen hintereinander. Des Weiteren existiert ein nachgelagerter Wartebereich für Taxen. Der Wartebereich hat 16 Reihen bzw. Spalten und 25 Zeilen. Initial stehen 16 Taxen in einer Zeile nebeneinander. Jede Reihe hat eine Ampel L01 bis L16 mit dem Status rot oder grün. Die Ampeln werden von einer zentralen Leitwarte über die Kommandos RED und GREEN gesteuert. Des Weiteren beobachtet die Leitwarte kontinuierlich die letzte Position der beiden Spuren. Der Wartebereich wird initial mit 400 Taxen befüllt. Zu Simulationszwecken wird eine Kette mit 400 zufälligen Zeichen aus dem Bereich [S,L] generiert. Für die Kunden existiert ein Schalter, über den die sukzessive Verteilung erfolgt. Die Simulation endet, wenn keine wartenden Kunden mehr existieren. Die Reihe im Wartebereich wird sukzessive von links nach rechts abgearbeitet. Initial stehen alle Ampeln auf rot. Für das initiale Befüllen des Taxistands werden für die zwei Spuren je vier Taxen benötigt. Die ersten acht Ampeln L01 bis L08 werden von rot auf grün geschaltet. Die ersten 4 Taxen ordnen sich sukzessive in die Spur I ein. Die verbleibenden vier Taxen fahren in Spur II. Die acht Ampeln L01 bis L08 werden wieder auf rot geschaltet und je ein Taxi in dieser Reihe rückt nach. Alle 500 ms wird ein Kunde dem Schalter entnommen und der zuständigen Spur bzw. dem Taxi zugeordnet. An einem Beispiel wird der Ablauf beschrieben. Der erste wartende Kunde ist vom Typ L und wird der Spur II zugeordnet. Der Kunde steigt in das Taxi und fährt ab. Die verbleibenden drei Taxen in der Spur rücken vor und die letzte Position wird frei. Das Kontrollcenter wird automatisch informiert und sendet das Kommando GREEN an Ampel L09. Das erste Taxi in dieser Reihe fährt auf die freigewordene Position, L09 wird wieder auf rot geschaltet und ein Taxi in dieser Reihe rückt nach.

Es werden die PrimzahlTupel (i) Primzahlzwilling – $(p, p+2)$, (ii) Primzahltrilling – $(p, p+2, p+6)$ oder $(p, p+4, p+6)$, (iii) Primzahlvierling – $(p, p+2, p+6, p+8)$, (iv) Primzahlfünfling – $(p, p+2, p+6, p+8, p+12)$ unterschieden. Für jeden Typ existiert ein separater Filter (i) twin, (ii) triplet, (iii) quadruple und (iv) quintuplet sowie ein eigenes Repository. Für den Bereich $n = [1..1000000]$ werden die Primzahlen (Anzahl: 78498) ermittelt und in einem Repository gespeichert. Für die Ermittlung der Primzahlen stehen die Strategien (i) Native und (ii) SieveEratosthenes zur Verfügung. Anschließend werden die Filter sukzessive angewandt und die Ergebnisse im korrespondierenden Repository zum Filter gespeichert. Beispielsweise befindet sich im Repository twin_repository der Datensatz mit dem Tupel (3,5). Nach dem Filtern wird je Repository eine statistische Analyse durchgeführt und ein Bericht (Builder) am Bildschirm ausgegeben. Der Bericht umfasst die Elemente (i) Datum / Uhrzeit, (ii) Anzahl der Tupel, (iii) Median der Abstände zwischen zwei Tupeln. Abstand: erster Wert vom zweiten Tupel minus zweiter Wert des ersten Tupel.

Ein Flughafen hat zwei Landebahnen R1 und R2. Für jede Landebahn existiert eine eigene Feuerwache. Bei den Flugzeugen werden die Typen A319, A320, A330, A350, A380, B737, B747, B777 und B787 unterschieden. Jede Feuerwache hat acht Einsatzfahrzeuge V01 bis V08. Bei einem Notfall werden in Abhängigkeit von dem Flugzeugtyp verschiedene Einsatzfahrzeuge beauftragt: (i) A319: V01, V02; (ii) A320: wie A319 + V03; (iii) A330: wie A320 + V04 und V05; (iv) A350: wie A320 + 2x V06; (v) A380: wie A350 + V07 + 2x V08; (vi) B737: wie A320; (vii) B747: wie A380; (viii) B777: wie A380 und (ix) B787: wie A350. Je Landebahn wird initial eine Liste mit 200 zufällig ausgewählten Flugzeugtypen generiert. Zu Simulationszwecken landet pro Sekunde je ein Flugzeug auf R1 und R2. Die Wahrscheinlichkeiten für einen Notfall betragen (i) A319 und B737: 0,05; (ii) A320 und B747: 0,03; (iii) A330 und B787: 0,02; (iv) A350, A380 und B777: 0,01. Zum Zeitpunkt der Landung wird entschieden, ob ein Notfall vorliegt. Bei einem Notfall drückt der Lotse im Tower den Alarmknopf und die Informationen (Flugzeug(e) und Bahn(en)) werden sofort an die Feuerwachen kommuniziert, da diese den Alarmknopf kontinuierlich beobachten. Basierend auf dem Regelwerk ermittelt die Feuerwache die benötigten Einsatzfahrzeuge. Die Einsatzfahrzeuge werden mit dem Kommando MOVEOUT und der Information zu Landebahn und Flugzeugtyp von der Leitwarte beauftragt. Die unterstützende Feuerwache bei den Flugzeugtypen A350, A380, B777 und B787 entsendet automatisch die zusätzlich benötigten Einsatzfahrzeuge. Der Einsatz dauert eine Sekunde und die Fahrzeuge stehen sofort wieder für den nächsten Einsatz zur Verfügung. In jeder Leitwarte existiert ein Button Disaster. Diese Button werden vom Tower kontinuierlich beobachtet. Sofern nicht ausreichend Einsatzfahrzeuge zur Verfügung stehen, drückt ein Mitarbeiter der Leitwarte – in der für die Landebahn zuständigen Feuerwache – diesen Button und der Tower wird automatisch informiert.

Ein Flugzeug bewegt sich mit konstant 2 pro Sekunde in einer Höhe von 100 und ist 500 von der Landebahn entfernt. Ab einer bestimmten Entfernung zur Landebahn (Descent Point) sinkt das Flugzeug konstant mit 3 Grad und landet punktgenau. Unter Berücksichtigung dieser Faktoren wird vom Tower der Descent Point berechnet. Der Tower beobachtet kontinuierlich das Flugzeug. Sobald das Flugzeug den Descent Point passiert, sendet der Tower das Kommando DESCENT an das Flugzeug und es wird der Sinkflug mit konstant 3 Grad eingeleitet. Hierzu werden die beiden Flaps je Tragfläche durch ein Kommando FLAPS3 vom Cockpit an die zentrale Kontrolleinheit (Mediator) eingestellt.

Ein Billigflieger bietet zwei Tickets (i) economy und (ii) premium an. Das Ticket economy kostet 50 Euro und bietet einen Stehplatz. Für das Ticket economy gilt die Kondition: 1x Handgepäck frei. Für das Ticket economy können die Zusatzleistungen (i) Check-In Gepäck bis 15 Kilo zu 30 Euro, (iii) Priority Boarding 10 Euro, (iv) Getränke-/Snackpaket zu 15 Euro. Das Getränke-/Snackpaket umfasst ein Softdrink und ein Hotdog aus der Mikrowelle. Das Ticket premium kostet 120 Euro und bietet einen Sitzplatz, ein Check-In Gepäck sowie Priority Boarding. Folgende Zusatzleistungen sind für beide Tickets verfügbar (i) Beitrag zu 15 Euro für zusätzliches Kerosin bei Warteschleifen, (ii) Frühstück mit einem Glas Sekt, einem Milchkaffee und einem Muffin zu 20 Euro. Die Tickets werden ausschließlich in elektronischer Form ausgestellt und an das Smartphone gesandt. Ein Ticket wird dargestellt als 2-dimensionale Matrix mit 10 Zeilen (y: 0-9) und 10 Spalten (x: 0-9). Zu Simulationszwecken werden die Tickets (i) economy + 1x Check-In Gepäck, (ii) economy + 2x Check-In Gepäck + 1x Getränke-/Snackpaket, (iii) premium + 1x zusätzliches Check-In Gepäck + 1x Beitrag zu Kerosinkosten + 1x Frühstück, (iv) premium + 1x Getränke-/Snackpaket + 1x Frühstück und (v) premium + 1x Frühstück. Das Ticket wird über einen Builder erstellt. Der obere und untere Rand wird als Kette mit Zeichen aus dem Bereich [0,1] dargestellt. Die linke Seite enthält die nächsthöhere Primzahl zu dem Dezimalwert des oberen Randbereichs. Die rechte Seite enthält die nächstniedrigere Primzahl zu dem Dezimalwert des unteren Randbereichs. Nicht benötigte Felder am linken und rechten Randbereich werden mit dem Zeichen x aufgefüllt. Ab der Zeile 2 und Spalte 2 werden folgende Elemente – getrennt durch * - als Fließtext angezeigt: (i) Datum, (ii) Flugnummer, (iii) Name, (iv) Geburtsdatum, (v) Klasse – economy oder premium, (vi) inkludierte Leistungen, (v) zusätzlich gebuchte Leistungen und (vi) Gesamtbetrag. Der linke und rechte Randbereich wird durch den Fließtext nicht überschrieben. Für das Boarding stehen zwei Schalter (i) normal und (ii) priority zur Verfügung. Zwecks Boarding werden die fünf spezifizierten Ticketkombinationen zufällig auf 150 Kunden verteilt. Die Kunden werden sukzessive durch einen Mitarbeiter (COR) auf beide Schalter verteilt.

Zielsetzungen

- a) Wiederholung und Vertiefung des Wissens zu Design Patterns.
- b) Praktische Anwendung der Design Patterns auf komplexe Aufgabenstellungen.
- c) Optimale Klausurvorbereitung im Hinblick auf eine vorzugsweise sehr gute Bewertung.

Wichtige Hinweise:

- **Pro Team** werden **zwei Aufgaben** bearbeitet.
- Die **Zuordnung** der **zwei Aufgaben** zu einem Team erfolgt **mit** einem **Zufallsgenerator**.
- **Bearbeitung** der Aufgaben **lokal** auf den Rechnern und Nutzung der Templates.
- **Studium** der **Struktur** und **Funktionsweise** der beteiligten **Design Patterns**.
- Verwendung geeigneter **englischer** Begriffe für **Namen** und **Bezeichnungen**.
- **Modellierung** von zwei Klassendiagrammen in Enterprise Architect.
Bitte
 - nutzen Sie das **Theme „DHBW“** (Visual Style → Visual Appearance → Diagram).
 - legen Sie für jede Aufgabe ein separates Klassendiagramm an.
 - benennen Sie das Klassendiagramm mit task<id>, z.B. task50.
 - benennen Sie die Datei mit <team_id>_<task_id>.eap, z.B. 03_50.eap.
 - **exportieren** Sie das Klassendiagramm als **PDF**-Datei mit einer A4-Seite im Querformat.
- **Implementierung** einer einwandfrei lauffähigen Applikation in Java 8.
Bitte
 - erstellen Sie ein **Paket** mit der Bezeichnung taskgroup<team_id>, z.B. taskgroup03.
 - erstellen Sie für jeden Task ein **Unterpaket** mit der Bezeichnung task<id>, z.B. task50.
 - nutzen Sie die **camelCase-Notation**, um die Lesbarkeit zu vereinfachen.
- **Test** der Implementierung mit **JUnit** und Gewährleistung der Funktionsweise.
- **Modellierung** und **Implementierung** wird mit **je 5 Punkten pro Aufgabe** bewertet.
- **Erstellung einer vollständigen und verschlüsselten 7-Zip-Datei unter Beachtung des Prozedere für die Abgabe von Prüfungsleistungen und der Namenskonvention.**
- **Zeitansatz:** 10 Stunden
- **Abgabetermin:** Sonntag, 18.02.2018
- **Bewertung:** Testat, 20 Punkte