Prof. Dr.-Ing. Peter Liggesmeyer
Dipl.-Inf. Max Steiner

TU Kaiserslautern
Dept. of Computer Science
AG Software Engineering: Dependability

Software Quality Assurance (WS12/13)

Problem Set 4

Due: in exercise, 12.12.2012

Problem 1: State-based Test

Given is the specification of a digital watch software.

For adjustment of a digital watch, the following states are to be considered:

- *Normal time:* State after inserting the battery
- *Adjust Hours:* Hours can be adjusted
- *Adjust Minutes:* Minutes can be adjusted
- *Adjust Seconds:* Seconds can be adjusted

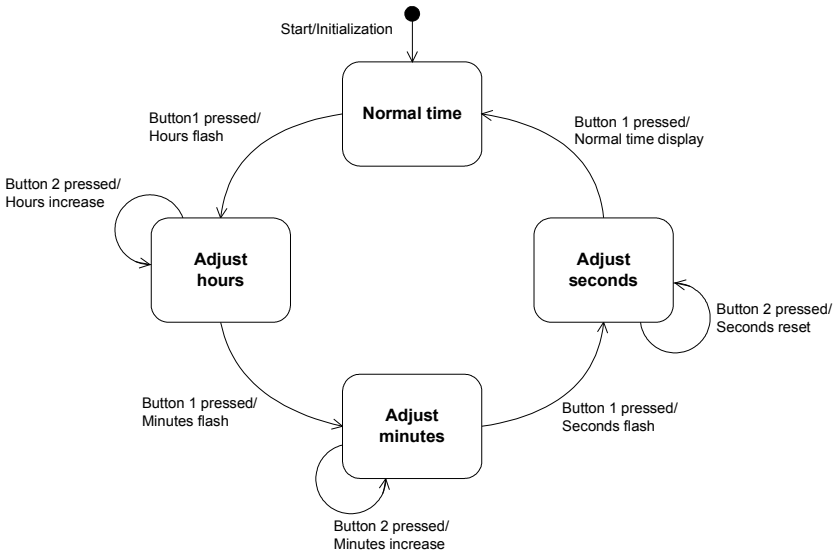
The following events could occur:

- *Start signal:* Battery inserted
- *Button 1 pressed*
- *Button 2 pressed*
- The two buttons must not be pressed simultaneously.

The following outputs could happen:

- *Hours flash:* The operator is currently in the hour editing mode.
- *Minutes flash:* The operator is currently in the minute editing mode.
- *Seconds flash:* The operator is currently in the second editing mode.
- *Hours increase:* The hour display has increased by 1 hour.
- *Minutes increase:* The minutes display increases by 1 minute.
- *Seconds reset:* 00 displays as second display.
- *Initialization:* Display of 00:00:00

State chart “Watch adjustment“



a) Please determine the test data for the program execution that traverses every state. Please select the simplest test cases.

b) Please determine the test data for the program execution that traverses every transition. Please select the simplest test cases.

Problem 2: Testing Object-oriented Software

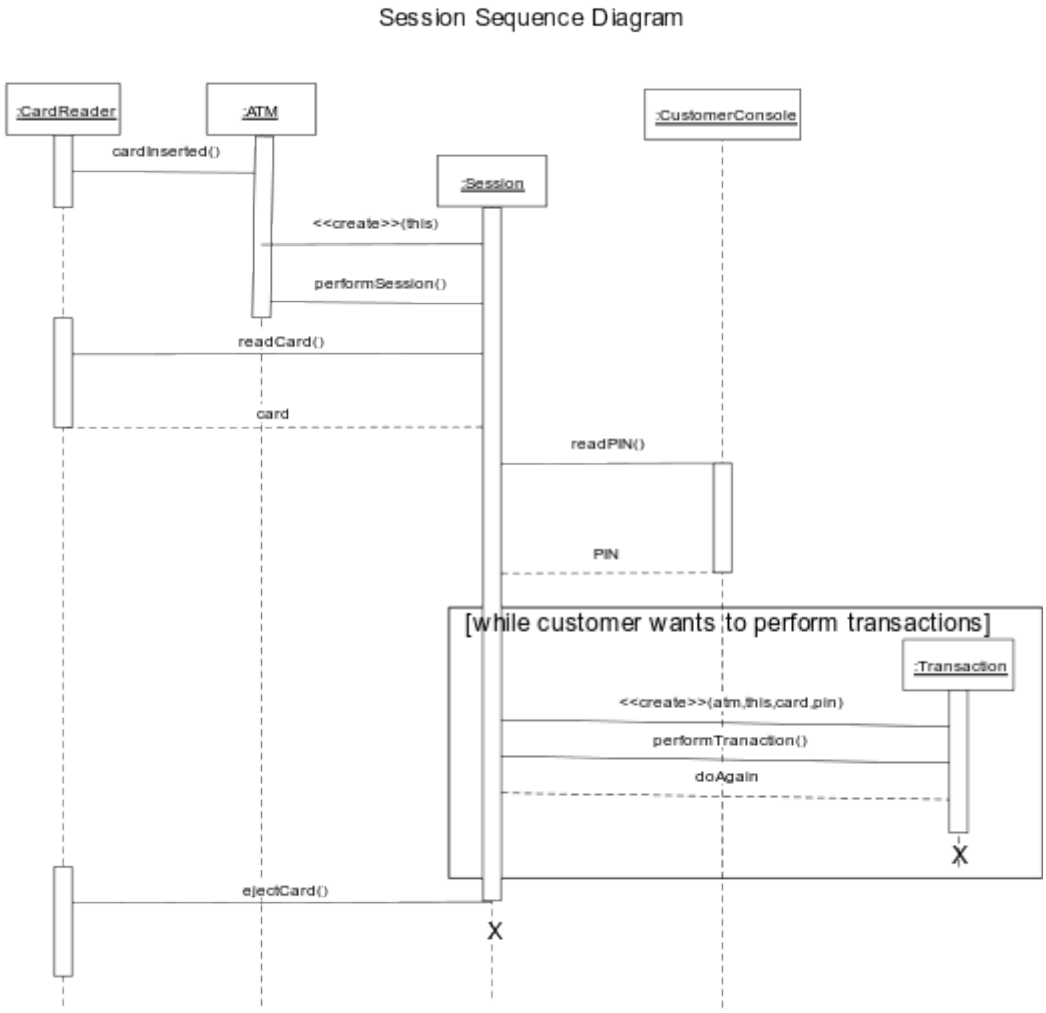
Given is a simplified specification of ATM operation sequences:

1. Customer enters card into the card reader
2. Customer console requesting pin entry is displayed
3. An interleaved sequence of digits is proved by the ATM
4. If user enters wrong pin three times, the card will be retained
5. Withdrawal transaction is created and performed
6. Cash dispenser dispenses approved amount to customers
7. Session ends if no transaction is active
8. Card is returned to customer
9. Possibility of cancellation by customer

The following additional requirements are to be considered:

1. PINs are four-digit number sequences, meaning ≥ 0001 and ≤ 9999
2. Withdrawal is only possible by choosing the menu options that are displayed by the customer console.
3. For this cash machine, it is only possible to input digits.

a) Please draw the **state transition diagram** for one session context according to this description. A session sequence diagram is shown below, which will additionally help you to derive the state chart.



b) Class Test (Unit Test)

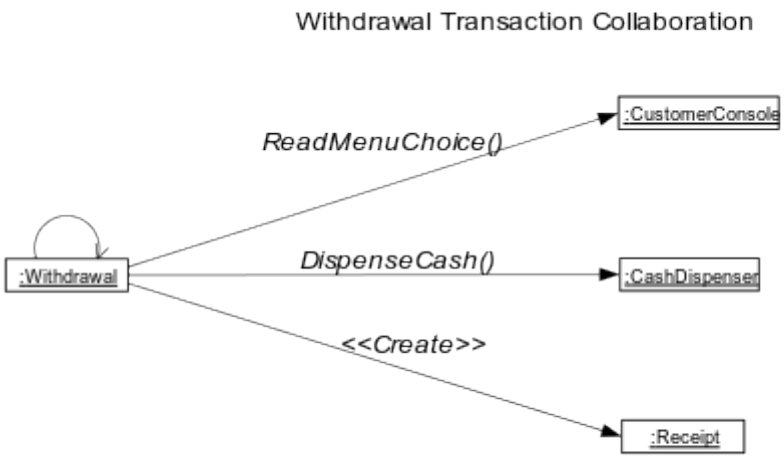
After an initial analysis, the following components (individual classes) are identified:

- Operator panel
- Card reader
- Customer console, consisting of a display and keyboard
- Network connection to the bank
- Cash dispenser
- Receipt printer
- Session
- Transaction

Attached you will find an excerpt of a standard for Test Documentation. Please read it and derive a unit test plan for this ATM system that includes the classes mentioned above.

c) Integration Test

Test of the intersection of the classes customerConsole and Withdrawal (sub-class of Transaction) via the message ReadMenuChoice() (see attached diagram Withdrawal Transaction Collaboration).



Interface specification of the operation `ReadMenuChoice()`

- The operation `ReadMenuChoice()` expects a non-negative value (10 value 500). It specifies the value of the bill in euros.
- The operation has the return value of the amount selected by the customer.

Interface parameter of the calling routine at the `CustomerConsole` device

- The following values can be chosen by the customer: `ReadMenuChoice()`: 20, 50, 100, 150, 200.

1. Please determine the assertion of this interface parameter for the integration test and derive the equivalence classes and test cases for input and output interfaces.
2. Please determine the equivalence classes and test cases for the operation `readPIN()` as well (with Boundary Value Analysis, `readPIN()` returns 4-digit number put in by the customer).
3. Attached you will find an excerpt of a standard for Test Documentation. Please read it and derive a integration test plan for this ATM system with an emphasis on the interfaces `ReadMenuChoice()` and `readPIN()`.

d) System Test

1. For a functional system test, what functions need to be tested? How do you ensure the completeness of your test set?
2. How do you test system performance (reaction time and the like)?

Now the customer of this ATM system demands to have an additional menu option for the user, the “300” option. This option is obviously not included in the available menu options, and should be added to both the CustomerConsole and the Withdrawal classes. The interface specification is altered, and ReadMenuChoice() is overwritten accordingly as well.

3. Do you need new test cases for this new menu option? Why? If so, which test cases do you need? Should the assertion be changed?
4. Attached you will find an excerpt of a standard for Test Documentation. Please read it and derive a system test plan with the consideration of performance aspect and the new menu option. Have you used your state chart? Where did you use it?