# Техніки тест дизайну при розробці тестових випадків

# Завдання

**Equivalence partitioning and Boundary value analysis**

To calculate the electricity cost user should enter an old and a new values of electricity counter and press “Calc” button. It works only if values are entered correctly:

- contain only digits;
- no longer than 6 digits;
- new value is not less than an old one;

1. Build equivalence classes (partitions) based on given information
2. Stand Out boundary values

# Завдання

**Decision tables**

E-store works with three types of customers according to the total sum of their orders: Regular, Silver and Gold. The list of products to be shown to the client is built according to the rules: products with label Silver or Gold are placed in the top of the list for corresponding types of clients. All the rest products are alphabetically sorted, except the promo products, which have the highest priority and are placed the topmost.

1. Build decision table based on given information.

**State transition**

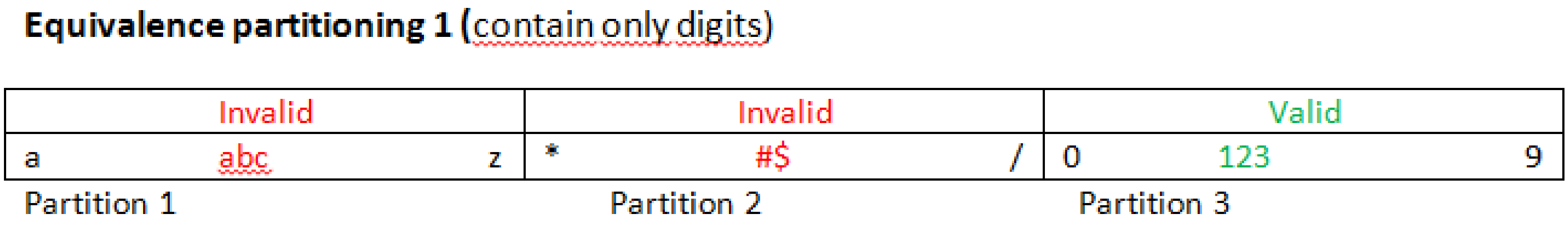
User wants to replenish the account using self-service kiosk. He enters the account number and in case it is active the user is asked to enter the sum and select the way of replenishment: from the credit card or by cash. After the cash/credit card is inserted the system checks whether enough money is available (available on the credit card or inserted into the kiosk). If there is enough money the account is replenished. In other case the user is asked to correct sum or cancel the operation.

1. Build state transition diagram based on given information

# План (Agenda)

- Equivalence partitioning
- Boundary value analysis
- Decision tables
- State transition

# Equivalence partitioning 1 (contain only digits)

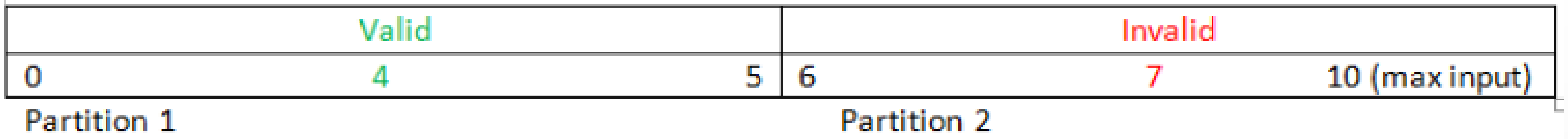


**Equivalence partitioning 1** (contain only digits)

| Invalid | Invalid | Valid |
|---|---|---|
| a abc z | * #$ / | 0 123 9 |

Partition 1 Partition 2 Partition 3

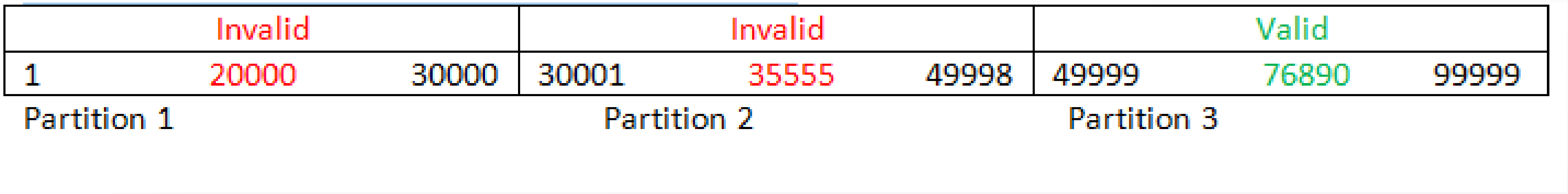
# Equivalence partitioning 2 (no longer than 6 digits)



| Valid | | | Invalid | | |
|---|---|---|---|---|---|
| 0 | 4 | 5 | 6 | 7 | 10 (max input) |
| Partition 1 | | | Partition 2 | | |

# Equivalence partitioning 3 (new value is not less than an old one)

- Let old value be 49999.
- The table with the values of new entered value:



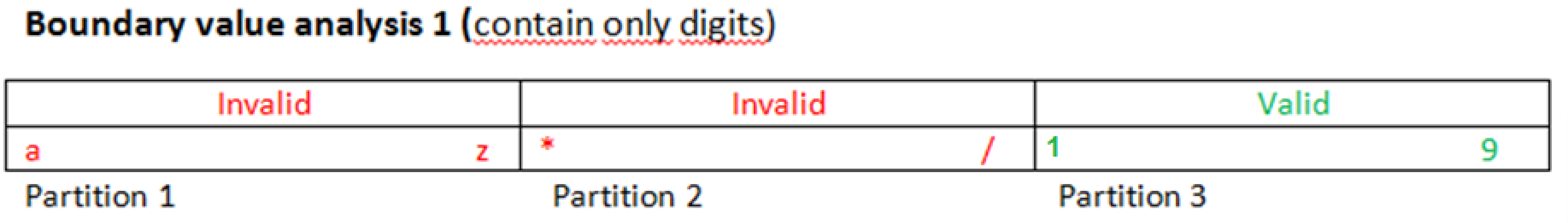
| Invalid | | | Invalid | | | Valid | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 20000 | 30000 | 30001 | 35555 | 49998 | 49999 | 76890 | 99999 |

Partition 1

Partition 2

Partition 3

# Boundary value analysis 1 (contain only digits)



**Boundary value analysis 1 (**contain only digits)

| Invalid | Invalid | Valid |
|---|---|---|
| a z | * / | 1 9 |

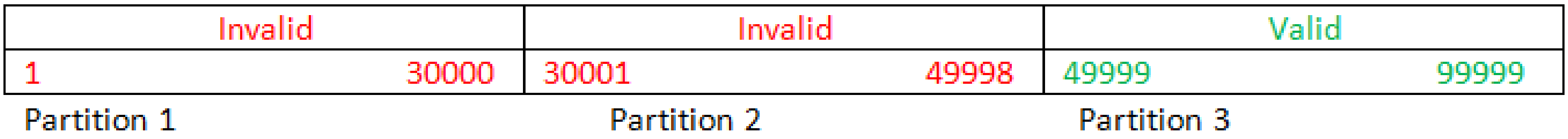
Partition 1 Partition 2 Partition 3

# Boundary value analysis 2 (no longer than 6 digits)

| Invalid | Invalid |
|---|---|
| 0 5 | 6 10 (max input) |
| Partition 1 | Partition 2 |

# Boundary value analysis 3 (new value is not less than an old one)

- Let old value be 49999.
- The table with the values of new entered value:



| Invalid | | Invalid | | Valid | |
|---|---|---|---|---|---|
| 1 | 30000 | 30001 | 49998 | 49999 | 99999 |
| Partition 1 | | Partition 2 | | Partition 3 | |

# Decision tables (1)

| Causes (inputs) | R1 | R2 | R3 | R4 | R5 | R6 | R7 | R8 |
|---|---|---|---|---|---|---|---|---|
| Regular User | + | + | + | + | - | - | - | - |
| Silver user | + | + | - | - | + | + | - | - |
| Gold user | + | - | + | - | + | - | + | - |

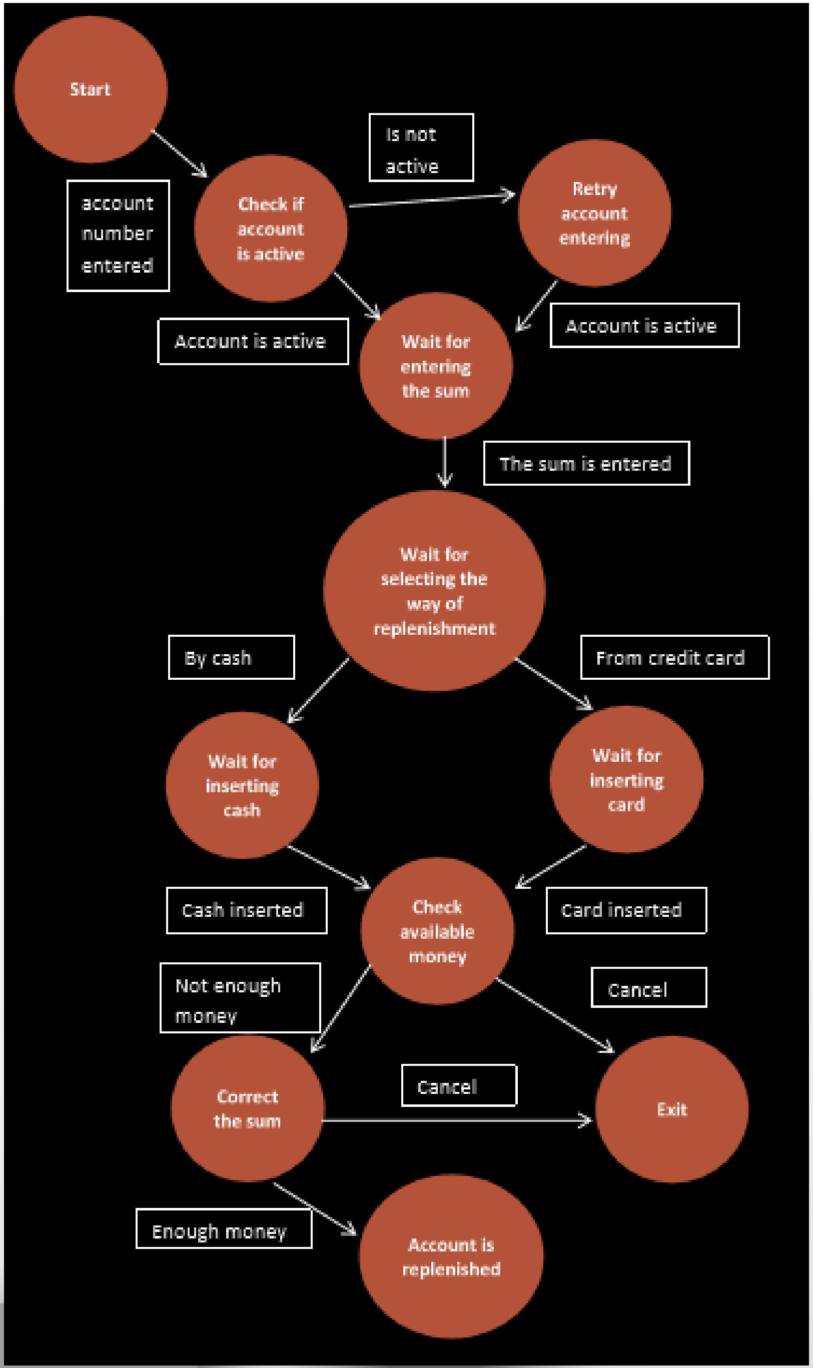
# Decision tables (2)

| Causes (inputs) | R4 | R6 | R7 | R8 |
|---|---|---|---|---|
| Regular User | + | - | - | - |
| Silver user | - | + | - | - |
| Gold user | - | - | + | - |

# Decision tables (2)

| **Causes (inputs)** | **R4** | **R6** | **R7** | **R7** |
|---|---|---|---|---|
| • Regular User | + | - | - | - |
| • Silver user | - | + | - | - |
| • Gold user | - | - | + | - |
| | | | | |
| **Effects (outputs)** | | | | |
| • Silver products on top | - | + | - | - |
| • Promo products (topmost) | + | + | + | + |
| • Rest products alphabetically sorted | - | + | + | - |
| • Gold products on the top | - | - | + | - |
| • Rest products alphabetically sorted on the top | + | - | - | + |

# State transition



# Висновок

Виконуючи дану лабораторну роботу, ми навчилися використовувати техніки тест дизайну для визначення набору тестів.