

[CSE2016]

프로그램 설계 방법론

Week 6: Control Structure : Conditional Statements

한양대학교 ERICA

소프트웨어융합대학

ICT 융합학부

2018. 10. 30



문제1. 홀수짝수 구분하기

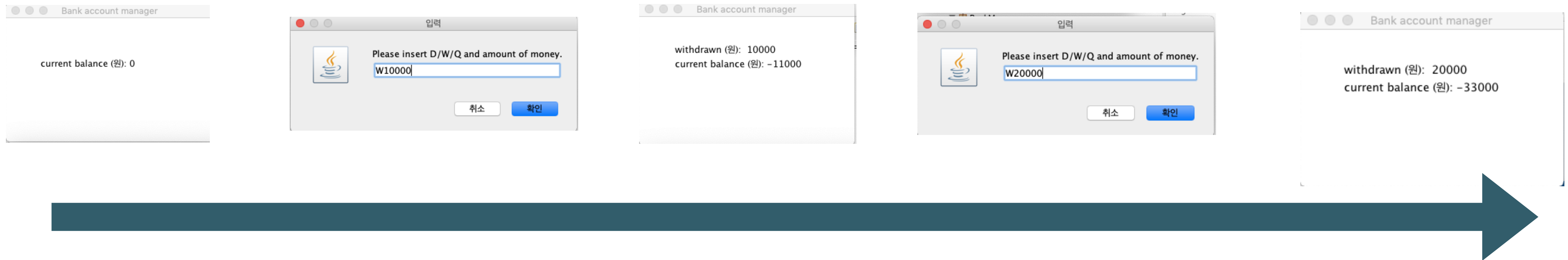
- switch 문을 사용하여, Scanner 를 이용해 입력 받은 숫자가 홀수인지 짝수인지 판별하세요.

```
<terminated> Switch [Java Applet]
숫자를 입력해주세요.5
홀수
```

문제2. 마이너스 통장

- 마이너스 통장이란 잔액이 마이너스로 빠지는 대출 상품입니다. 수업자료의 Bank Manager Case Study를 수정한 코드를 수정하여 (뒷장에 첨부), 아래의 조건을 만족하는 ‘마이너스 통장’ 기능을 하는 애플리케이션을 완성하세요.
- 인출하려는 금액이 잔액보다 많을 경우, 잔액을 초과하는 인출금은 대출금으로 간주합니다.
- 대출금은 10%의 이자가 붙어서 대출됩니다. (단, 이자의 소수점은 무시)
- 애플리케이션 구조 : 수업자료 (Week06) 28쪽에서 확인
- 뒷쪽에 첨부된 코드는 수업자료의 Bank Manager Case Study 를 달러(\$) 에서 원화(W)로 바꾸기만 한 예시입니다.

문제2. 실행예시



- [예시] 잔고가 0원인데 1만원을 빌리는 경우, 최종 잔고는 - 1만 1천원.
- 그 후 다시 2만원을 빌려 갔을 때, 최종 잔고는 - 3만 3천원 (=1만 1천원 + 2 만 2천원).

문제2. 팁

- Bank Manager Case 에서 활용한 MVC 모델을 이해하면서 수업자료에 제시된 대로 코드를 작성합니다.
- (1) 어떤 부분에서 현재 잔액을 확인하고, (2) 조건문을 어떻게 활용하여 어떻게 대출금에 10%의 이자를 추가해야할지 단계적으로 생각하여 작성하면됩니다.

문제2. 기본 코드 (AccountController.java)

```
public class AccountController {
    private BankReader reader;
    private BankWriter writer;
    private BankAccount account;

    public AccountController(BankReader r, BankWriter w, BankAccount a) {
        reader = r;
        account = a;
        writer = w;
    }

    public void processTransactions() {
        char command = reader.readCommand("Please insert D/W/Q and amount of money.");
        switch (command) {
            case 'Q':
                System.exit(0);

            case 'D': {
                int amount = reader.readAmount();
                if (account.deposit(amount))
                    writer.showTransaction("deposit (원): ", amount);
                else
                    writer.showTransaction("deposit error: ", amount);
                break;
            }
            case 'W': {
                int amount = reader.readAmount();
                if (account.withdraw(amount))
                    writer.showTransaction("withdrawn (원): ", amount);
                else
                    writer.showTransaction("withdraw error: ", amount);
                break;
            }
            default:
                writer.showTransaction("invalid input: " + command);
        }
        this.processTransactions();
    }
}
```

문제2. 기본 코드 (AccountManager.java)

```
public class AccountManager {  
  
    public static void main(String[] args) {  
        BankReader reader = new BankReader();  
        BankAccount account = new BankAccount(0);  
        BankWriter writer = new BankWriter("Bank account manager", account);  
        AccountController controller = new AccountController(reader, writer, account);  
        controller.processTransactions();  
    }  
  
}
```

문제2. 기본 코드 (BankAccount.java)

```
import javax.swing.JOptionPane;

public class BankAccount {
    private int balance;

    public BankAccount(int initial_amount) {
        if (initial_amount >= 0) {
            balance = initial_amount;
        } else {
            balance = 0;
        }
    }

    public int getBalance() {
        return balance;
    }

    public int getInterest(int exceed) {
        return (int) (exceed * 0.1);
    }

    public boolean deposit(int amount) {
        boolean result = false;
        if (amount < 0)
            JOptionPane.showMessageDialog(null, "invalid input.");
        else {
            balance = balance + amount;
            result = true;
        }
        return result;
    }

    public boolean withdraw(int amount) {
        boolean result = false;
        if (amount < 0)
            JOptionPane.showMessageDialog(null, "invalid input.");
        else if (amount > balance)
            JOptionPane.showMessageDialog(null, "not enough balance.");
        else {
            balance = balance - amount;
            result = true;
        }
        return result;
    }
}
```


문제2. 기본 코드 (BankReader.java)

```
package BankManager_copy;

import javax.swing.JOptionPane;

public class BankReader {
    private String input_line = "";

    public char readCommand(String message) {
        input_line = JOptionPane.showInputDialog(message).toUpperCase();
        return input_line.charAt(0);
    }

    public int readAmount() {
        int amount = 0;
        String s = input_line.substring(1, input_line.length());
        if (s.length() > 0) {
            amount = new Integer(s).intValue();
        } else
            JOptionPane.showMessageDialog(null, "input for amount is not provided.");

        return amount;
    }
}
```

문제2. 기본 코드 (BankWriter.java)

```
package BankManager_copy;

import java.awt.*;
import javax.swing.*;
import java.text.*;

public class BankWriter extends JPanel {
    private int WIDTH = 300;
    private int HEIGHT = 200;
    private BankAccount bank;
    private String last_transaction = "";

    public BankWriter(String title, BankAccount b) {
        bank = b;
        JFrame f = new JFrame();
        f.getContentPane().add(this);
        f.setTitle(title);
        f.setSize(WIDTH, HEIGHT);
        f.setBackground(Color.white);
        f.setVisible(true);
    }

    public void showTransaction(String message, int amount) {
        last_transaction = message + " " + amount;
        this.repaint();
    }

    public void showTransaction(String message) {
        last_transaction = message;
        this.repaint();
    }

    public void paintComponent(Graphics g) {
        g.setColor(Color.white);
        g.fillRect(0, 0, WIDTH, HEIGHT);
        g.setColor(Color.black);
        int text_margin = 50;
        int text_baseline = 50;
        g.drawString(last_transaction, text_margin, text_baseline);
        g.drawString("current balance (원): " + bank.getBalance(), text_margin, text_baseline + 20);
    }
}
```