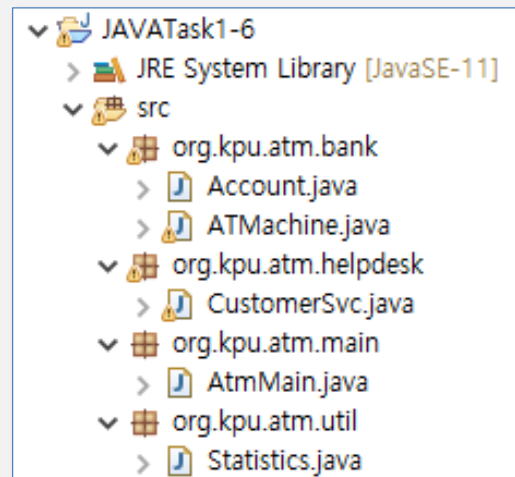


1. 과제 내용 : ATM 프로그램 작성
2. 수행 방법 : 개인 프로그램 작성  
프로젝트 명: **영문이니셜\_ATM** (예: **gildong\_ATM**)
3. 제출 항목 : 압축한 프로젝트 디렉토리와 jar 파일을 제출  
압축 파일 명: **이름\_ATM.zip** 과 **영문이니셜\_ATM.jar**
4. 제출 기한 : **10월 8일까지**
5. 기능 설명 (**추가 기능은 순서대로 구현할 것**)



기능(배점)	내용	자바 클래스
기본(4점)	계좌 개설 기능, 계좌 조회 기능, 업무 종료 기능	
추가(4점)	1. 계좌 입금 기능	AtmMain.java, ATMachine.java Account.java
	2. 계좌 출금 기능	
	3. 계좌 이체 기능	
	4. 고객 센터 기능 (암호 변경)	CustomerSvc.java
	5. 고객 관리 기능 (통계 기능) - ATM 현금 잔고, 고객 잔고 총액 - 고객 잔고 평균, 고객 잔고 최고 - 고객 계좌 현황 ( <b>잔고 내림차순 정렬</b> )	Statistics.java ( <b>static 매소드 구현</b> )

## 기본 : AtmMain.java 와 결과 화면

```

public class AtmMain {
    public static void main(String[] args) {
        // 초기 설정 - 개설가능한 최대계좌수(1000계좌),
        // ATM 잔금(50만원), 관리자 암호
        ATMachine atm=new ATMachine(1000, 500000, "admin");

        Scanner scan = new Scanner(System.in);
        while(true) {
            atm.displayMenu();
            System.out.println(" 메뉴를 선택하세요 >> ");
            try {
                int select = scan.nextInt();
                switch(select) {
                    case 1: // 계좌 개설
                        atm.createAccount();
                        break;
                        .....
                }
            } catch (InputMismatchException e) {
                System.out.println(" 정확하게 입력해주세요.");
                continue;
            }
        }
    }
}

```

```

-----
-          KPU bank          -
-----
1. 계좌 개설
2. 계좌 조회
9. 업무 종료
메뉴를 선택하세요 >> 1
-----개설-----
이름 입력: 김미영
암호 입력: kim
김미영님 100번 계좌번호가 정상적으로 개설되었습니다. 감사합니다

-----
-          KPU bank          -
-----
1. 계좌 개설
2. 계좌 조회
9. 업무 종료
메뉴를 선택하세요 >> 2
-----조회-----
계좌번호 입력: 100
비밀번호 입력: kim
계좌 잔액 : 0

-----
-          KPU bank          -
-----
1. 계좌 개설
2. 계좌 조회
9. 업무 종료
메뉴를 선택하세요 >> 9
안녕히 가세요 !

```

## Account.java

```
package org.kpu.atm.bank;

public class Account {
    private int nID;           // 계좌 번호
    private int nBalance;      // 계좌 잔고
    private String strAccountName; // 고객 명
    private String strPassword; // 계좌 비밀번호

    public Account(int id, int money, String name, String password) { // 생성자
    }

    boolean authenticate(int id, String passwd) { // 계정 확인
    }

    public int getnID() {
        return nID;
    }

    public int getnBalance() {
        return nBalance;
    }
}
```

## ATMachine.java

```
package org.kpu.atm.bank;

import java.util.Scanner;

public class ATMachine {
    private Account [] accountArray;    // 고객계좌배열 참조변수
    private int nMachineBalance;        // ATM 잔고
    private int nMaxAccountNum;        // 고객계좌 참조변수 배열크기
    private int nCurrentAccountNum;    // 개설된 고객계좌 수
    private String strManagerPassword; // 관리자 비밀번호

    public static final int BASE_ACCOUNT_ID = 100; // 고객 계좌 발급시 시작 번호

    public ATMachine(int size, int balance, String password) { // 생성자
    }

    public void createAccount() { // 계좌 개설
    }

    public void checkMoney() { // 계좌 조회
    }

    public void displayMenu() { // 메인 메뉴 표시
    }
}
```

## 추가 기능 1 : 계좌 입금

Account.java

```
int deposit(int money)    // 입금
```

ATMachine.java

```
public void depositMoney() // 계좌 입금
```

```
-----
-      KPU bank      -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
9. 업무 종료
메뉴를 선택하세요 >> 1
-----
      개설
-----
이름 입력: 김미영
암호 입력: kim
김미영님 100번 계좌번호가 정상적으로 개설되었습니다. 감사합니다

-      KPU bank      -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
9. 업무 종료
메뉴를 선택하세요 >> 3
-----
      입금
-----
계좌번호 입력: 100
비밀번호 입력: kim
입금 액 입력: 6000
입금 후 잔액: 6000

-----
-      KPU bank      -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
9. 업무 종료
메뉴를 선택하세요 >> 9
안녕히 가세요 !
```

## 추가 기능 2 : 계좌 출금

Account.java

```
public int widraw(int money) // 출금
```

ATMachine.java

```
public void widrawMoney() // 계좌 출금
```

```
-----
-      KPU bank      -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
9. 업무 종료
메뉴를 선택하세요 >> 1
-----
      개설
-----
이름 입력: 김미영
암호 입력: kim
김미영님 100번 계좌번호가 정상적으로 개설되었습니다. 감사합니다
-----
-      KPU bank      -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
9. 업무 종료
메뉴를 선택하세요 >> 3
-----
      입금
-----
계좌번호 입력: 100
비밀번호 입력: kim
입금액 입력: 6000
입금 후 잔액: 6000
```

```
-----
-      KPU bank      -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
9. 업무 종료
메뉴를 선택하세요 >> 4
-----
      출금
-----
계좌번호 입력: 100
비밀번호 입력: kim
출금 액 입력: 4000
출금 후 잔액: 2000
-----
-      KPU bank      -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
9. 업무 종료
메뉴를 선택하세요 >> 9
안녕히 가세요 !
```

## 추가 기능 3 : 계좌 이체

ATMachine.java

public void transfer() // 계좌 이체

```

-----
-      KPU bank      -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
9. 업무 종료
메뉴를 선택하세요 >> 1
-----개설-----
이름 입력: 김미영
암호 입력: kim
김미영님 100번 계좌번호가 정상적으로 개설되었습니다. 감사합니다
-----
-      KPU bank      -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
9. 업무 종료
메뉴를 선택하세요 >> 3
-----입금-----
계좌번호 입력: 100
비밀번호 입력: kim
입금액 입력: 9000
입금 후 잔액: 9000

```

```

-----
-      KPU bank      -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
9. 업무 종료
메뉴를 선택하세요 >> 1
-----개설-----
이름 입력: 박철호
암호 입력: park
박철호님 101번 계좌번호가 정상적으로 개설되었습니다. 감사합니다
-----
-      KPU bank      -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
9. 업무 종료
메뉴를 선택하세요 >> 5
-----이체-----
계좌번호 입력: 100
비밀번호 입력: kim
이체계좌 입력: 101
이체금액 입력: 4000
이체 계좌를 다시 확인하세요.
현재 잔액: 5000
계좌 이체를 완료하였습니다.

```

## 추가 기능 4 : 고객 센터 (비밀번호 변경)

CustomerSvc.java

```
public CustomerSvc(Account [] acctArray, int nCurrentAcctNum) // 생성자
public void updatePasswdReq() // 비밀번호 변경
```

ATMachine.java

```
// 비밀번호 변경요청
public void requestSvc ()
```

Account.java

```
// 비밀번호 변경
public boolean updatePasswd(String
oldPasswd, String newPasswd)
```

```
-----
- KPU bank -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
6. 고객 센터
9. 업무 종료
메뉴를 선택하세요 >> 1
-----계좌-----
이름 입력: 박철호
암호 입력: park
박철호님 100번 계좌번호가 정상적으로 개설되었습니다. 감사합니다
-----
- KPU bank -
-----
1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
6. 고객 센터
9. 업무 종료
메뉴를 선택하세요 >> 6
-----암호변경-----
계좌번호 입력: 100
기존 비밀번호 입력: park
신규 비밀번호 입력: cheolho
비밀번호를 수정하였습니다.
```



## 추가 기능 5 : 고객 관리

Account.java

```
public String getAccountName() // 고객 명 읽기
```

ATMachine.java

```
public void managerMode() // 고객 관리
```

Statistics.java

```
public static int sum(Account [] account, int size)
public static double average(Account [] account, int size)
public static int max(Account [] account, int size)
public static Account [] sort(Account [] account, int size)
```

## 추가 기능 5 : 고객 관리

- KPU bank -

1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
6. 고객 센터
7. 고객 관리
9. 업무 종료

메뉴를 선택하세요 &gt;&gt; 1

-----개설-----

이름 입력: 김미영

암호 입력: kim

김미영님 100번 계좌번호가 정상적으로 개설되었습니다. 감사합니다

- KPU bank -

1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
6. 고객 센터
7. 고객 관리
9. 업무 종료

메뉴를 선택하세요 &gt;&gt; 3

-----입금-----

계좌번호 입력: 100

비밀번호 입력: kim

입금액 입력: 12000

입금 후 잔액: 12000

- KPU bank -

1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
6. 고객 센터
7. 고객 관리
9. 업무 종료

메뉴를 선택하세요 &gt;&gt; 1

-----개설-----

이름 입력: 박철호

암호 입력: park

박철호님 101번 계좌번호가 정상적으로 개설되었습니다. 감사합니다

- KPU bank -

1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
6. 고객 센터
7. 고객 관리
9. 업무 종료

메뉴를 선택하세요 &gt;&gt; 3

-----입금-----

계좌번호 입력: 101

비밀번호 입력: park

입금액 입력: 8000

입금 후 잔액: 8000

- KPU bank -

1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
6. 고객 센터
7. 고객 관리
9. 업무 종료

메뉴를 선택하세요 &gt;&gt; 7

-----고객관리-----

관리자 비밀번호 입력: admin

ATM 현금 잔고: 520000

고객 잔고 총액: 200000원(2명)

고객 잔고 평균: 10000원

고객 잔고 최고: 12000원

고객 계좌 현황(고객 잔고 내림 차순 정렬)

김미영 100 12000원

박철호 101 8000원

- KPU bank -

1. 계좌 개설
2. 계좌 조회
3. 계좌 입금
4. 계좌 출금
5. 계좌 이체
6. 고객 센터
7. 고객 관리
9. 업무 종료

메뉴를 선택하세요 &gt;&gt;