

JAVA 프로그래밍 과제 보고서

학 과	정보통신융합공학부
학 번	1862007
성 명	김보람
작 품 명	지켜줄게 너의 알바비
작품 설명	보통 시급으로 급여를 받는 알바생들은 자신들이 한 일에 대한 정확한 급여를 받지 못한다. 그 점을 개선하고 싶어서 프로그램을 구상하게 되었고 알바생의 출근과 퇴근 시간을 계산하여 정확한 급여를 구해주는 프로그램을 만들게 되었다. 또한 관리자는 손쉽게 급여를 지급할 수 있도록 도와주는 프로그램이다.

위와 같이 JAVA 프로그램 과제 보고서를 제출합니다.

2019 년 6 월 19 일

제 출 자

김 보 람

(서명)

1. 프로그램 명

지켜줄게 너의 알바비

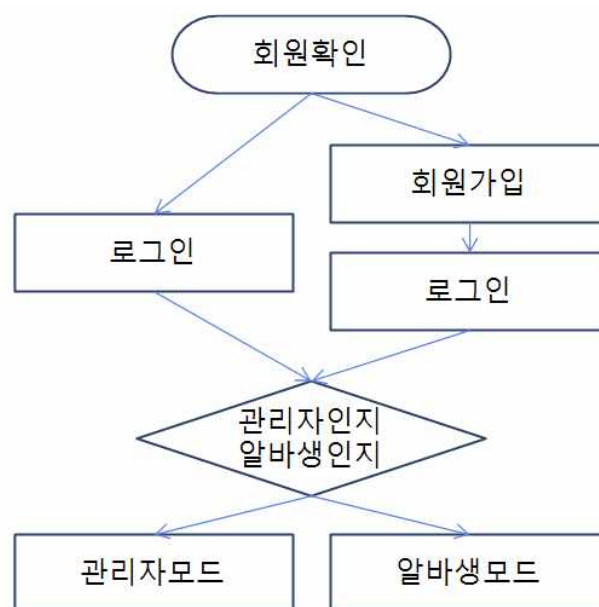
2. 프로그램의 목표

이 프로그램의 목표는 시급을 받는 알바생들의 정확한 급여를 구해주는 것이다. 알바생들은 1시간을 일할 때와 1시간 30분을 일할 때 모두 같은 급여를 받는 경우가 빈번하다. 이 프로그램을 통해 이러한 경우를 해결하여 자신의 노동력에 대한 정확한 댓가를 받을 수 있을 것이다. 또한 관리자도 알바생에게 급여 지급을 쉽게 하도록 한다.

3. 프로그램 내용

프로그램이 시작되면 파일 입출력을 이용하여 사용자가 회원가입과 로그인한다. 회원가입시 저장된 관리자인지 알바생인지의 정보에 따라 관리자모드와 알바생모드로 프로그램이 진행된다. 관리자 모드에서는 알바생들의 한달 급여와 알바생의 이름, 계좌, 계좌의 은행을 볼 수 있다. 알바생모드에선 자신의 출퇴근 시간에 따른 정확한 급여를 알 수 있고 자신의 하루 급여를 알 수 있다. 관리자 모드에선 알바생의 총 급여와 알바생의 계좌, 계좌의 은행을 알 수 있도록 하여 쉽게 지급하도록 도와준다.

4. 프로그램 구조도



5. 프로그램 소스 및 실행화면

<회원가입 후 로그인>

지켜줄게 너의 알바비

☆☆☆☆환영합니다☆☆☆☆
회원이신가요?
1.네 2.아니요 회원이 되고싶어요!

2

☆☆☆☆회원가입을 시작합니다☆☆☆☆

이름 : 알바생보람
ID : dkfqk
password : dkfqkpw
확인을 위해 다시 한번 입력해주세요...
password : dkfqk123
password가 서로 다릅니다. 다시 한번 입력해주세요

dkfqkpw
관리자라면 1을 입력하고 알바생이라면 2를 입력하세요
2
급여받을 계좌번호는?1234566789
그 계좌는 어느 은행 계좌인가요?국민은행

회원가입되었습니다!
로그인을 진행해 주세요

☆☆☆☆로그인을 시작합니다☆☆☆☆

아이디 :

이 화면은 회원가입을 진행하는 화면이다. 우선 사용자에게 회원여부를 입력받는다. 회원이 아닐 경우 회원가입을 진행한다. 회원가입시 입력받는 기본 정보는 ID, PASSWORD, 관리자여부이다. 만약 회원가입하는 사람이 알바생일 경우 추가적으로 계좌번호와 계좌의 은행정보를 입력받는다. PASSSSWORD를 입력받을 때 2번을 입력받아 같은 PASSWORD일 경우에만 회원가입이 진행된다. 회원가입이 끝나면 자동으로 로그인이 시작된다.

- 소스 중 중요부분

1. 입력한 것을 읽기위해 생성

```
BufferedReader reader = new BufferedReader(new InputStreamReader(System.in));
```

2.joiner라는 객체를 생성하고 그 객체에 정보를 저장하여 읽는다

```
System.out.print("이름 : ");  
joiner.name = reader.readLine();
```

3.비밀번호를 2번 입력받고 equals를 사용하여 두 비밀번호를 비교하고 while문을 사용하여 비밀번호가 다르면 갈때까지 입력받는다

```
while(joiner.PASSWORD2.equals(joiner.PASSWORD1)==false)  
{  
    System.out.println("password가 서로 다릅니다. 다시 한번 입력해주세요 \n");  
    joiner.PASSWORD2 = reader.readLine();  
}
```

4.가입자가 관리자인지 알바생인지에 따라 추가정보 요청이 달라진다. 관리자는 null값을 지정한다

```
if(master == 1)  
{  
    joiner.account="NULL";  
    joiner.bank="NULL";  
}
```

```

else if(master == 2)
{
    System.out.print("급여받을 계좌번호는?");
    joiner.account = reader.readLine();
    System.out.print("그 계좌는 어느 은행 계좌인가요?");
    joiner.bank = reader.readLine();
}
else //잘못 입력했을 경우 오류창
{
    System.out.println(" ");
    System.out.println("잘못 입력하셨습니다");
    System.out.println("다시 입력해주세요");
    System.out.println(" ");
}

```

5.do-while 문을 이용하여 master값에 1,2만 입력받을 수 있도록 한다

```

do
{
    ~생략~
}
}while(master!=1 && master!=2);

```

6. 입력한 정보를 저장한다.

```

writer.write(String.format("%s", joiner.name, joiner.ID, joiner.PASSWORD1, joiner.account,
joiner.bank, master));

```

<로그인>

<div>지켜줄게 너의 알바비</div> <div> <p>☆☆☆☆환영합니다☆☆☆☆</p> <p>회원이신가요?</p> <p>1. 네 2. 아니요 회원이 되고싶어요!</p> </div> <div>1</div> <div> <p>☆☆☆☆로그인을 시작합니다☆☆☆☆</p> <p>아이디 : master</p> <p>비밀번호 : masterpw</p> </div> <div>로그인 성공!</div> <div>이름 : 보람 님 안녕하세요!</div>	<p>이 화면은 로그인을 진행하는 화면이다. 사용자가 입력한 ID, PASSWORD를 변수에 저장한다. 지정된 경로에 있는 파일에 파일에 저장된 ID와 PASSWORD를 비교한다. ID와 PASSWORD가 모두 같을 경우 로그인에 성공한다. 로그인에 성공하면 사용자의 이름을 불러와 이름과 안녕하세요라는 문장을 출력한다.</p>
---	---

-소스중 중요부분

1.파일에서 스트림을 통해 읽어들인다

```

input = new BufferedReader(new FileReader("c:\\Temp\\member.txt"));

```

2. 부분별 주석으로 설명

```
while ((s = _input.readLine()) != null)
{
    String[] split = s.split(" "); //한 줄 가져와, 스플릿 한다. 파싱 기준은 공백인 \t\t
    //id와 password에는 이미 입력받은 값이 저장되어 있으므로,
    //텍스트 파일에서 read한 값을 다시 덮어씌우면 안된다.

    //id = split[1];
    //password = split[2];
    // 버퍼를 다 빼 내었기에 버퍼리더를 닫아 준다.
    if(id.equals(split[1]) && password.equals(split[2]))
    { //입력한 값과 저장된 값 비교하고 아이디 비밀번호가 모두 같으면 로그인 성공
        _input.close(); // BufferedReader 멈춤
        System.out.println(" [로그인 성공!]");
        System.out.println(" [로그인 성공!]");
        System.out.println(" [로그인 성공!]");
        loginSuccess = true; //로그인 성공여부
        names = split[2]; //저장한 이름 변수 지정
        System.out.println(" [이름 : " + split[0] + " 님 안녕하세요!]");
        System.out.println(" [이름 : " + split[0] + " 님 안녕하세요!]");

        System.out.println(" [관리자 여부 확인]");

        master = Integer.parseInt(split[5]);
        //관리자여부 return 해주어야 하기 때문에 불러와서 master 변수에 저장
    }
}
```

<알바생 모드>

아이디 : java
비밀번호 : javapw

로그인 성공!

이름 : 자바 님 안녕하세요!

< 메뉴 >
1. 출근했어요!
2. 종료할래요

1
출근한 시간 : 13시26분35초

< 메뉴 >
퇴근시 enter키를 눌러주세요

퇴근한 시간 : 13시57분36초
일한시간 : 00:31:01
오늘은 4311원을 벌었어요

이 화면은 알바생이 프로그램을 사용하는 화면이다. 로그인한 사람의 정보(master: 관리자여부 확인하는 변수)를 불러와 알바생일 경우 이 화면을 출력한다. '1. 출근했어요'를 선택하면 선택했을 때의 시간을 불러오고 퇴근때까지 stopwatch가 작동한다. 퇴근 시 enter를 누르면 stopwatch가 멈추고 enter를 누른 퇴근때의 시간을 출력해주고 일한 시간과 하루 급여를 출력해준다.

1. 출퇴근 선택시 시간을 출력해주기 위해 calendar를 사용
함부로 접근하지 못하도록 private선언

2.출근 후 stopwatch가 작동하며 퇴근하면 stopwatch의 작동이 멈추고 시간과 시급을 곱하여 하루 급여(today_money)를 출력한다.

```
int today_money = Money.today_money(); //money에서 계산한 하루급여를 today_money에 저장
System.out.println("오늘은 " + today_money+"원을 벌었어요");
```

이 화면은 관리자가 프로그램을 사용하는 화면이다. 로그인한 사람의 정보(master: 관리자여부 확인하는 변수)를 불러와 관리자일 경우 이 화면을 출력한다. 알바생의 일급을 다 더한 급여를 호출하며 그 알바생의 계좌번호와 계좌의 은행정보를 불러와 관리자가 급여지급을 하기 쉽게 해준다.

1. 알바생의 이름, 급여, 계좌번호, 은행 정보 호출

```
while ((s = reader2.readLine()) != null)
{
    String[] split = s.split(" ");

    if(split[1] != null )
    {
        System.out.println(" 알바생 "+split[0]+"의 이번달 월급 : "+split[5]+"원 입니다");
        System.out.println("계좌번호는 "+split[3]+"이며 은행은 "+split[4]+"은행 입니다");
    }
}
```

<StopWatch 클래스 설명>

1.타이머 메소드 : 1을 입력받으면 타이머 시작 0을 입력받으면 타이머 종료

```
static void stopwatch(int onOff) {
    if (onOff == 1) // 타이머 켜기
    {
        oldTime = (int) System.currentTimeMillis() / 1000;
    }

    else if (onOff == 0) // 타이머 끄고, 시분초를 timerBuffer 에 저장
    {
        secToHHMMSS( ((int) System.currentTimeMillis() / 1000) - oldTime );
    }
}
```

2. secToHHMMSS 메소드 : 정수로 된 시간을 초단위(sec)로 입력 받아, "04:11:15" 등의 형식의 문자열로 시분초를 저장

```
static void secToHHMMSS(int secs) {
    sec = secs % 60;
    min = secs / 60 % 60;
    hour = secs / 3600;

    timerBuffer = String.format("%02d:%02d:%02d", hour, min, sec);
}
```

3. pause 메소드 : 타이머 종료 전까지 기다림

```
static void pause() {
    try {
        System.in.read();
    } catch (IOException e) { }
}
```

<Money 클래스 설명>

1. pay 변수 : 시간당 급여, 초기값은 최저시급인 8350원으로 설정

```
static int pay = 8350; //초기값 : 최저시급
```

2. today_money메소드 : 타이머의 시간을 불러와 사용자의 하루급여를 계산하고 return해줌

```
static int today_money(){

    int today_money ;

    int hour = Stopwatch.hour;
    int min = Stopwatch.min;
    int sec = Stopwatch.sec;

    today_money = hour * pay + min*(pay / 60) + sec*(pay/60/60);

    return today_money;}
}
```

<memeber 메모장>

member - 메모장									
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)									
마스터보람		master		masterpw		NULL		NULL	1
자바	java		javapw	123-123-123		국민		2	4311

<master 메모장>

Temp > 1862007_김보람		master - 메모장	
이름		파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)	
master		java	4311