------------------------------

1. 파일명 : CertifiMachineRun

public class CertifiMachineRun // 메인메소드 포함 클래스

{

public static void main(String[] args) {}

}

-----------------------------------

2. 파일명 : FirstScreen

public class FirstScreen // 초기 화면 -> 메뉴 선택 까지

{

void fsPrint() // 초기 안내문구 출력 메소드

{}

void fsSelectMenu() // 재학생, 졸업생, 관리자 메뉴 선택 메소드

{

// 인스턴스 변수명

// 1. 재학생 rg

// 2. 졸업생 gr

// 3. 관리자 ad

// 선택한 옵션의 로그인 메소드로 이동

}

}

-----------------------------------

3. 파일명 : StudentMenu ( →RegiStudent으로 바꿔야대 왜냐면 파일이름은 public class의 클래스명!!!!!)

public interface StudentMenu //– 다른 파일에서 이 인터페이스 가져갈거아니니까 public 붙일 필요 없음

.

public class RegiStudent implements StudentMenu // 학생메뉴

{

스캐너 인스턴스 생성

StList클래스 인스턴스 생성

String putHak, putPw;

static int certType, su; // 다른 클래스에서도 쓸거니까 static 선언.

// 그리고 static 변수를 다루는 메소드도 static 해줘야함.

@Override

public void login()

{

loginConfirm(); 메소드 호출

}

@Override

public void loginConfirm()

{

로그인 성공하면

selectType(); 호출

로그인 실패하면

login(); 호출

}

@Override

public void selectType() // 증명서 선택 메소드드 (static void selectType()으로 고쳐야댐)

{

선택한 증명서의 출력 메소드 호출

}

@Override

public void a4Num() // 선택한 증명서의 인쇄 매수 받는 메소드 (static void a4Num() )

{

callBasket(); 메소드 호출

}

@Override

public void callBasket()

{

장바구니에 담으시겠습니까?

Y → BaksketandPay클래스의 addElement1(); 호출

N → selectType(); 호출

}

public void tuitionPayment()

{

a4Num(); 호출

}

public void transcript()

{

a4Num(); 호출

}

public void enrollment()

{

a4Num(); 호출

}

}

class GraphicsDrawLineEx

class GradStudent

{

@Override

public void selectType()

{

transcript();

// 또는

gradscript();

}

@Override

public void a4Num()

{

callBasket();

}

public void gradscript() // ②졸업증명서

{

a4Num();

}

}

-----------------------------------

4. 장바구니와 결제

class Menus // 변수만 생성

class BasketandPay

{

public void addElement1()

{

menuSelect();

}

public void menuSelect()

{

menuRun();

}

public void menuRun()

{

HowtoPay();

BacktoMenu()

delBasket();

}

public void HowtoPay()

{

card();

cash();

account();

}

public void BacktoMenu()

{

rg.selectType();

}

public void delBasket()

{

rg.selectType();

}

void card()

{}

void cash()

{}

void account()

{}

}

class Loading

{

**public static void loading()**

**{}**

}

class BasketandPay2

{

public void addElement2()

{

menuSelect2();

}

public void menuSelect2()

{

menuRun2();

}

public void menuRun2()

{

HowtoPay2();

BacktoMenu2()

delBasket2();

}

public void HowtoPay2()

{

card();

cash();

account();

}

public void BacktoMenu2()

{

gr.selectType();

}

public void delBasket2()

{

gr.selectType();

}

void card()

{}

void cash()

{}

void account()

{}

}

-----------------------------------

5. AdminMenu // 관리자메뉴

public class AdminMenu

{

// ① (관리자 로그인) 비밀번호 입력 메소드 (완료)

void adLogin()

{

성공 :adSelect();

실패 : adLogin()

}

void adSelect()

{

adIncome();

adManage();

adPaper();

adStManage();

adExit();

adClose();

}

void adIncome()

()

void adManage()

{

관리고르면 adCashFull(); 호출

}

void adCashFull() {}

void adPaper() {

재고관리 고르면 adPpFull(); 호출

}

void adPpFull() {}

void adStManage()

{

adStManageBackMenu();

}

void adStManageBackMenu()

{

돌아가기 : adStManage();

또는

관리자메뉴 : adSelect();

}

void adExit() {}

void adClose() {}

}

—-------------------------------------------------

6. StList

public class StList

{

public void stList() {}

public void readList() {}

public Map<String, String> print() {}

public void putNew() {}

public void change() {}

public void delete() {}

}