|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2019\_1\_java1\_12** | **학번 :** |  | **이름 :** |  |

* **내용 점검**

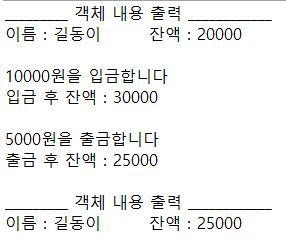
|  |
| --- |
| 1. 생성자 오버로딩, 접근제한자, getter & setter 메소드 |
| **class** Student{  //필드 선언  **private** String n; //이름을 저장하기 위한 필드 선언  **private** **int** j, p; //과목을 저장하기 위한 필드 선언    //디폴트 생성자 - 생성자 오버로딩  Student(){  **this**("", 0, 0); //오버로딩 된 생성자 호출, 첫번째 실행문으로 작성해야함  }    //생성자 오버로딩  Student(String n, **int** j, **int** p){ //매개변수로 받은 값을 필드 값으로 초기화  **this**.n=n;  **this**.j=j;  **this**.p=p;  }    //getter method : 이름 필드 값 반환  //메소드 실행 후 문자열 반환, 따라서 메소드 자료형은 String  //반환값은 return 키워드 옆에 작성. 단, 한번에 한개의 값만 반환  **public** String getN() {  **return** n;  }  //setter method : 이름 필드 값 변경  **public** **void** setN(String n) {  **this**.n = n;  }  //메소드 구현  **int** sum(){ //과목 합을 계산하여 반환하는 메소드  **return** j+p;  }    **double** ave() {  **return** sum()/2;  }    **public** String toString() { //객체 내용을 문자열로 반환하는 메소드  String str="이름 : " +n +"\t과목1:"+j + "\t과목2:"+p;  str += "\n합 : "+sum() + "\t평균 : " + ave();  **return** str;  }  }  //실행용 클래스 - main() 메소드를 제공  **public** **class** StudentTest {  **public** **static** **void** main(String[] args) {  //클래스의 필드와 메소드는 객체를 통해서 접근 해야 함  Student obj=**new** Student("김낙준", 90, 85); //Student 클래스 객체 obj 선언 & 생성  System.***out***.println("\_\_\_\_\_\_\_\_\_\_\_\_\_\_이름 변경 전 객체 내용\_\_\_\_\_\_\_\_\_\_\_");  System.***out***.println(obj); //toString() 메소드 호출  System.***out***.println("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_이름 변경 후 객체 내용\_\_\_\_\_\_\_\_\_\_\_");  obj.setN("김효근"); //obj 객체의 이름을 변경  System.***out***.println(obj); //toString() 메소드 호출  }  } |
| **[실행 결과]** |

* **프로그램 과제**

1. Student 클래스 과목 필드에 대한 getter와 setter 메소드를 추가하고, 추가된 코드를 제시하시오. 실행 결과는 제시하지 않아도 됨

|  |
| --- |
| **[프로그램 소스]** |
| **[실행결과]** |

1. 은행 계좌를 처리하기 위한 클래스를 작성하고 테스트하는 프로그램이다. 밑줄 친 부분을 채워 넣은 후 실행 결과를 제시하시오



**//은행 계좌 처리를 위한 클래스**

**public** **class** Account {

**//이름을 저장하기 위한 필드 name 선언, private**

**//잔액을 저장하기 위한 필드 balance 선언(4byte 정수형), private**

**//생성자 – 매개변수로 받은 값으로 이름과 잔액을 초기화- 본인작성**

**//형식 매개변수로 받은 값만큼 잔액을 증가하는 메소드, 반환값 없음 - 본인작성**

**void** dePosit(**int** money){

}

**//형식 매개변수로 받은 값만큼 잔액을 감소하는 메소드, 반환값 없음 – 본인작성**

**void** withDraw(**int** money){

}

**//필드 balance 값을 반환하는 getter 메소드**

**public** String toString(){ **//객체 내용(이름과 잔액)을 문자열로 반환**

}

}

**//Account 클래스에 대한 객체를 생성하고 테스트 하는 클래스**

**public** **class** AccountTest {

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

**//Account 클래스 객체 obj 선언 & 생성 – 본인작성**

**//dePosit()메소드 호출, 10000을 매개변수로 전달- 본인작성**

System.*out*.println("입금 후 잔액 : " + obj.getBalance());

/**/withDraw() 메소드 호출, 5000을 매개변수로 전달- 본인작성**

System.*out*.println("출금 후 잔액 : " + obj.getBalance());

System.*out*.println(obj); **//객체 내용 출력**

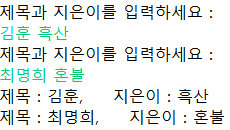
}

|  |
| --- |
| **[프로그램 소스]** |
| **[실행 결과]** |

1. 책을 나타내는 Book 클래스를 정의하시오.

|  |  |
| --- | --- |
| * 필드 : | 제목, 저자, private |
| * 메소드 | * 생성자 : 매개변수로 받은 값을 필드값으로 초기화 * 각 필드에 대한 getter, setter 메소드 * 객체 내용을 문자열로 반환하는 toString() 메소드 |

BookTest 클래스를 작성하여 제시된 결과처럼 실행되도록 한다.



|  |
| --- |
| **[프로그램 소스]** |
| **[실행 결과]** |

1. 제품 관리를 위한 Product 클래스를 작성하시오

* 필드 구성 : 제품번호(String, private), 제품명(String, private), 판매한 제품 갯수(int, private), 제품가격(int, private)
* 매개변수로 받은 값으로 필드를 초기화 하는 생성자와 모든 필드를 0으로 하는 디폴트 생성자
* 각 필드에 대한 접근자와 설정자 메소드
* getInvoice() 메소드 : 제품 판매액(제품가격 \* 판매한 제품 갯수)을 계산하여 int로 반환
* display() 메소드: 필드값과 제품 판매액을 출력, 반환값 없음

ProductTest 클래스를 작성하시오

* Prodect 객체 생성 시 필드 초기값을 생성자 매개변수로 전달. 단, 표준 입력 장치로 입력 받은 값을 생성자 매개변수로 전달.
* display()메소드를 호출하여 필드값과 제품 판매액 출력



|  |
| --- |
| **[프로그램 소스]** |
| **[실행 결과]** |