|  |  |  |
| --- | --- | --- |
| 2019\_2\_java1\_12 | 학번 : 20195124 | 이름 : 김민석 |

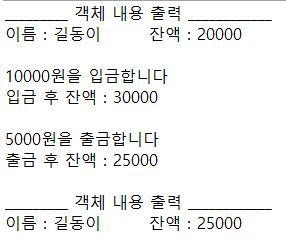
* **실행 결과 제출**

|  |
| --- |
| 1. 생성자 오버로딩, 접근 제한자, 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]** |

|  |
| --- |
| 1. 객체 배열 |
| **import** java.util.\*;  **class** Member{  **private** String id; //필드 선언- 정보은닉  **private** **int** pass;    **public** Member() { //디폴트 생성자  **this**(" ", 0); //오버로딩 된 생성자 호출  }    **public** Member(String id, **int** pass) { //생성자 오버로딩  **this**.id=id;  **this**.pass = pass;  }    **public** String toString() { //객체 내용 출력  **return** "id = " + id +"\tpassword = " + pass;  }  //getter 메소드 - 필드값 반환  **public** String getId() {  **return** id;  }  **public** **int** getPass() {  **return** pass;  }    //setter 메소드 - 필드값 변경  **public** **void** setId(String id) {  **this**.id = id;  }  **public** **void** setPass(**int** pass) {  **this**.pass = pass;  }  }  **public** **class** MemberTest {  **public** **static** **void** main(String[] args) {  Scanner key=**new** Scanner(System.***in***);  **int** i;  Member[] obj = **new** Member[5]; //Member 객체 배열 선언 & 생성    **for**(i=0; i<obj.length;i++) {  System.***out***.print("id를 입력하세요 : ");  //생성된 객체를 배열에 저장  obj[i] = **new** Member(key.next(), (**int**)(Math.*random*()\*9000)+1000); }    //객체 배열 원소 출력  **for**(Member m : obj)  System.***out***.println(m.toString());    System.***out***.print(("변경하기 전 id를 입력하세요 >>>>> "));  String id=key.next();  **for**( i=0; i<obj.length;i++) {  **if**(id.equals(obj[i].getId())) { //접근자 메소드를 이용하여 동일한 id를 검색  System.***out***.print("입력한 id로 변경합니다. id를 입력하세요 : ");  //입력한 id를 설정자 메소드로 전달하여 id 필드값 변경  obj[i].setId(key.next()); System.***out***.println("변경된 결과를 출력합니다");  System.***out***.println(obj[i].toString());  **break**;  }  }  **if**(i >= obj.length)  System.***out***.println("해당 id가 존재하지 않습니다");  }  } |
| **[실행 결과]** |

* **프로그램 과제**

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**

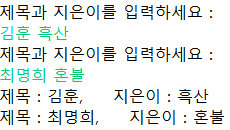
}

|  |
| --- |
| **[프로그램 소스]**  **//다른 개발환경이 이상해서 이클립스 사용했습니다..**  class Account {  private String name;  private int balance;    Account(String name, int balance) {  this.name = name;  this.balance = balance;  }    void dePosit(int money) { balance += money; }  void withDraw(int money) { balance -= money; }  public int getBalance() { return balance; }    public String toString() {  return "name : " + name + "\tbalance : " + balance;  }  }  public class J1\_1115\_HW1 {  public static void main(String[] args) {  Account obj = new Account("Kim", 10000);    obj.dePosit(10000);  System.***out***.println("입금 후 잔액 : " + obj.getBalance());    obj.withDraw(5000);  System.***out***.println("출금 후 잔액 : " + obj.getBalance());    System.***out***.println(obj);  }  } |
| **[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]**  import java.util.Scanner;  class Book {  private String title;  private String name;    public Book() { this(" ", " "); }  public Book(String title, String name) {  this.title = title;  this.name = name;  }    public void setTitle(String title) { this.title = title; }  public String getTitle() { return title; }  public void setName(String name) { this.name = name;}  public String getName() { return name; }    public String toString() {  return "제목 : " + title + ",\t지은이 : " + name;  }    }  public class J1\_1115\_HW2 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.***in***);  String name, title;  String[] str;  Book[] b = new Book[2];    for(int i = 0; i < b.length; i++) {  System.***out***.println("제목과 지은이를 입력하세요 : ");  name = sc.nextLine();  str = name.split(" ");  title = str[0];  name = str[1];  b[i] = new Book(title, name);  }  for(Book temp : b) {  System.***out***.println(temp);  }  }  } |
| **[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

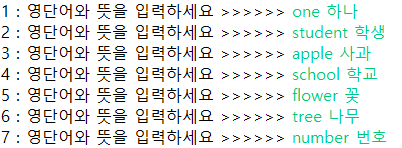
ProductTest 클래스를 작성하시오

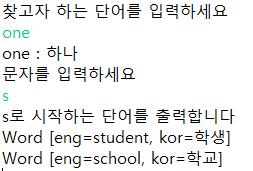
* Prodect 객체 생성 시 필드 초기값을 생성자 매개변수로 전달. 단, 표준 입력 장치로 입력 받은 값을 생성자 매개변수로 전달.
* display()메소드를 호출하여 필드값과 제품 판매액 출력



|  |
| --- |
| **[프로그램 소스]**  import java.util.Scanner;  class Product {  private String num;  private String name;  private int count;  private int price;    Product() { this(" ", " ", 0, 0); }  Product(String num, String name, int count, int price) {  this.num = num;  this.name = name;  this.count = count;  this.price = price;  }    public void setNum(String num) { this.num = num; }  public String getNum() { return num; }  public void setName(String name) { this.name = name; }  public String getName() { return name; }  public void setCount(int count) { this.count = count; }  public int getCount() { return count; }  public void setPrice(int price) { this.price = price; }  public int getPrice() { return price; }    int getInvoice() { return (price\*count); }    void display() {  String str = "num : " + num + "\tname : " + name + "\tcount : " + count;  str += "\tprice : " + price + "\tinvoice : " + this.getInvoice();  System.***out***.println(str);  }  }  public class J1\_1115\_HW3 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.***in***);  String[] str;  String name;    System.***out***.println("제품번호, 제품명, 갯수, 가격 입력 : ");  name = sc.nextLine();  str = name.split(" ");  Product p = new Product(str[0], str[1], Integer.*parseInt*(str[2]), Integer.*parseInt*(str[3]));  p.display();  }  } |
| **[실행 결과]** |

1. 키보드로 영 단어를 입력하면 단어 뜻을 한글로 출력하는 프로그램을 작성하시오. 영 단어와 한글은 객체 배열 생성시 초기화 하도록 한다.





class Word{

//영어 단어 저장을 위한 필드, private

//영 단어에 대한 뜻을 저장하는 필드, private

//생성자 – 매개변수로 받은 값을 필드로 초기화

//모든 필드에 대한 getter 메소드

//객체 내용을 문자열로 반환하는 toString() 메소드

}

class WordTest{

public static void main(String[] args){

//객체 배열을 선언 & 생성, 배열 크기는 7- 본인작성

//입력한 값으로 배열 초기화 - 본인 작성

//접근자 메소드를 사용하여 객체 배열에 저장된 영 단어와 입력된 영 단어를 equals() 메소드를 사용하여 검색, 일치하는 영단어를 찾으면 그에 해당하는 뜻을 출력 - 본인작성

//입력된 문자로 시작하는 영 단어와 뜻을 한글로 출력, charAt() 사용- 본인작성

**}**

**}**

|  |
| --- |
| **[프로그램 소스]**  import java.util.Scanner;  class Word {  private String name;  private String value;    public Word() { this(" ", " "); }  public Word(String name, String value) {  this.name = name;  this.value = value;  }    public String getName() { return name; }  public String getValue() { return value; }    public String toString() {  return "Word [eng = " + name + ", kor = " + value + "]";  }  }  public class J1\_1115\_HW4 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.***in***);  Word[] w = new Word[7];  String[] str;  String name, find;    for(int i = 0; i < w.length; i++) {  System.***out***.print((i+1) + " : 영단어와 뜻을 입력하세요 >>>>>>> ");  name = sc.nextLine();  str = name.split(" ");  w[i] = new Word(str[0], str[1]);  }    System.***out***.println("찾고자 하는 단어를 입력하세요");  find = sc.next();  for(Word temp : w) {  if(find.equals(temp.getName())) {  System.***out***.println(find + " : " + temp.getValue());  }  }    System.***out***.println("문자를 입력하세요");  find = sc.next();  System.***out***.println(find.charAt(0) + "로 시작하는 단어를 출력합니다.");  for(Word temp : w) {  if(find.charAt(0) == temp.getName().charAt(0)) {  System.***out***.println(temp);  }  }  }  } |
| **[실행 결과]** |