|  |  |  |
| --- | --- | --- |
| 2019\_2\_java1\_13 | 학번 : 20195124 | 이름 : 김민석 |

* **실행 결과 제출**

|  |
| --- |
| 1. 정적 메소드와 정적 필드  * 정적 메소드와 정적필드는 클래스 이름으로 호출 * 메소드 나 필드 선언 시 static 키워드 사용 * 정적 메소드에서는 정적 필드와 메소드만 사용하거나 호출할 수 있다 |
| **class** Compute{  **static** **int** add(**int** a, **int** b) {  **return** a+b;  }  **static** **double** add(**double** a, **double** b) {  **return** a+b;  }    **static** **int** add(**int**... a) { //가변 길이 매개 변수  **int** sum=0;  **for**(**int** t : a)  sum+=t;  **return** sum;  }  **static** **double** add(**double**[] dim) {  **double** sum=0;  **for**(**double** t : dim)  sum+=t;  **return** sum;  }  }  **public** **class** MethodOverloading {  **public** **static** **void** main(String[] args) {  **double** sum = Compute.*add*(34.2, 45.32);  System.***out***.println("sum(double) = " + sum);  System.***out***.println("sum(int) = " + Compute.*add*(34, 50));  System.***out***.println("sum(int...) = " + Compute.*add*(2, 4, 7, 12, 4, 6));  //무명배열을 매개변수로 전달  System.***out***.println("sum(double[]) = " + Compute.*add*(**new** **double**[] {3.4, 6.5, 2.4, 7.4, 12.8, 9.12})); |
| **[실행 결과]** |

|  |
| --- |
| 1. 싱글 톤 객체 – 프로그램 실행 중 한번만 객체 생성 |
| **class** Write{  **private** **static** Write *write* = **new** Write();  **private** Write() { }  **public** **static** Write getInstance() {  **return** *write*;  }  **void** print(**char**... a) { //가변 길이 매개 변수  **for**(**int** i=0; i<a.length;i++) {  System.***out***.println(a[(**int**)(Math.*random*()\*a.length)]);  }  }  }  **public** **class** SingleToneTest {  **public** **static** **void** main(String[] args) {  Write w=Write.*getInstance*();  w.print(**new** **char**[] { '◇','▷','▽','※','◎'});  }  } |
| **[실행 결과]** |

|  |
| --- |
| 1. 상속 |
| **class** Person {  **private** String name, tel; //이름과 연락처  **public** Person(String name, String tel) {  **this**.name = name;  **this**.tel = tel;  }  **void** disPlay() {  System.***out***.print("name=" + name + ", 연락처=" + tel);  }  }  **class** Member **extends** Person{  **private** String id; //아이디  **public** Member(String name, String tel, String id) {  **super**(name, tel); //부모 클래스 생성자 호출  **this**.id=id;  }  **void** disPlay() {  **super**.disPlay(); //부모 클래스 disPlay() 메소드 호출  System.***out***.println(", 아이디 : " + id);  }  }  **class** Student **extends** Person{  **private** **int** hak;  **public** Student(String name, String tel, **int** hak) {  **super**(name, tel);  **this**.hak=hak;  }  **void** disPlay() {  **super**.disPlay(); //부모 클래스 disPlay() 메소드 호출  System.***out***.println(", 학번 : " + hak);  }  }  **public** **class** InheritanceTest {  **public** **static** **void** main(String[] args) {  Member member=**new** Member("한림", "231-3421","java");  Student student = **new** Student("소중", "543-2341", 20191120);  member.disPlay();  student.disPlay();  }  } |
| **[실행 결과]** |

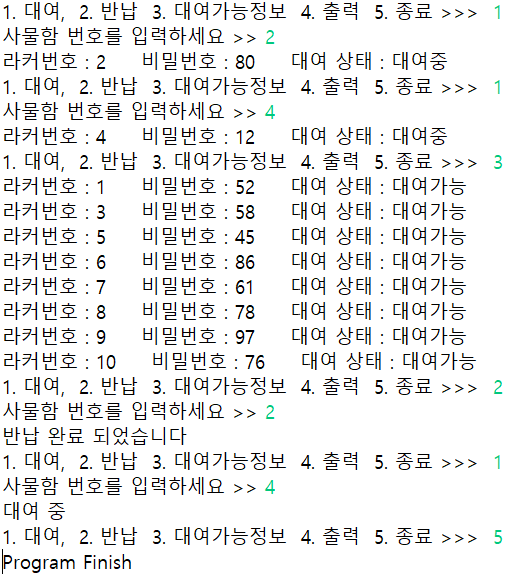
* **프로그램 과제**

1. 제시된 main() 메소드와 실행 결과를 참고하여 프로그램을 완성하시오

|  |  |
| --- | --- |
|  | **public** **class** Answer1 {  **public** **static** **void** main(String[] args) {  System.***out***.println("Max1 = " + Oper.*Max*(3.4, 5.6, 21.3, 5.2, 76.3, 34.5));  System.***out***.println("Max2 = " + Oper.*Max*(12.2, 5.4, 7.8, 11.3, 9.7, 35.4, 67.4,66.3, 56.1));  }  } |

|  |
| --- |
| **[프로그램 소스]**  class Oper {  public static double Max(double... d) {  double m = 0.0;    for(double temp : d) {  if(m <= temp) m = temp;  }  return m;  }  }  public class J1\_1122\_HW1 {  public static void main(String[] args) {  System.***out***.println("Max1 = " + Oper.*Max*(3.4, 5.6, 21.3, 5.2, 76.3, 34.5));  System.***out***.println("Max2 = " + Oper.*Max*(12.2, 5.4, 7.8, 11.3, 9.7, 35.4, 67.4,66.3, 56.1));  }  } |
| **[실행 결과]** |

1. 사물함을 관리하기 위한 프로그램을 완성 하시오.



**class** Lock{

//필드 선언(전용 멤버), 사물함 번호, 사물함 비밀번호, 대여상태(Boolean)

//생성자 – 대여 상태는 false로 할 것(대여 가능:false, 대여 중:true)

//비밀번호 필드에 대한 설정자

//사물함 번호에 대한 접근자

//대여 상태를 반환하는 isState()

//대여 상태를 변경하는 setState()

//객체 내용을 문자열로 반환하는 toString() – 출력결과 참고

}

**public** **class** Answer2 {

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

Lock[] lock=**new** Lock[10];

Scanner key=**new** Scanner(System.***in***);

**boolean** flag=**true**;

**int** num;

**for**(**int** i=0; i<lock.length;i++) { //객체 배열 초기화

lock[i]=**new** Lock(i+1,(**int**)(Math.*random*()\*90)+10);

}

**while**(flag) {

System.***out***.print("1. 대여, 2. 반납 3. 대여가능정보 4. 출력 5. 종료 >>> ");

**switch**(key.nextInt()) {

**case** 1:

System.***out***.print("사물함 번호를 입력하세요 >> ");

num=key.nextInt();

**if**(**Check.*state***(lock[num-1])) // 대여 상태 점검

System.***out***.println("대여 중");

**else** {

**Check.change**(lock[num-1]);

System.***out***.println(lock[num-1]);

}

**break**;

**case** 2:

System.***out***.print("사물함 번호를 입력하세요 >> ");

num=key.nextInt();

**if**(!**Check.state\_ch**(lock[num-1])) //대여 상태 변경

System.***out***.println("반납 완료 되었습니다");

**break**;

**case** 3:

**Check.printState**(lock);

**break**;

**case** 4:

**Check.printAll**(lock);

**break**;

**case** 5 :

flag=**false**;

}

}

System.***out***.println("Program Finish");

}

}

|  |
| --- |
| **[프로그램 소스]**  import java.util.Scanner;  class Lock {  private int num;  private int pass;  private boolean onBorrow;    public Lock(int num, int pass) {  this.num = num;  this.pass = pass;  onBorrow = false;  }    public void setPass(int newPass) { pass = newPass; }  public int getNum() { return num; }  public boolean isState() { return onBorrow; }  public void setState(boolean onBorrow) { this.onBorrow = onBorrow; }    public String toString() {  String str = "라커번호 : " + num + "\t비밀번호 : " + pass + "\t대여 상태 : ";  if(onBorrow) str += "대여중";  else str += "대여 가능";  return str;  }  }  class Check {  public static boolean state(Lock l) {  if(l.isState()) return true;  else return false;  }  public static void change(Lock l) {  l.setState(true);  }  public static boolean state\_ch(Lock l) {  l.setState(false);  return false;  }  public static void printState(Lock[] l) {  for(Lock temp : l) {  if(!temp.isState()) {  System.***out***.println(temp);  }  }  }  public static void printAll(Lock[] l) {  for(Lock temp : l) {  System.***out***.println(temp);  }  }  } |
| **[실행 결과]** |

1. 일반적인 음식을 나타내는 Food 클래스를 상속받아서 멜론을 나타내는 Melon 클래스를 작성하여 보자. Food 클래스는 칼로리, 가격, 중량 등의 정보를 가진다. Melon 클래스는 추가로 경작 농원 정보를 가진다. 생성자, 접근자, 설정자를 포함하여서 각각의 클래스를 작성한다. 이들 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하라.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | --- | | Food | | -cal : int  -cost : int  -kg : int | | +getCal() : int  +getCost() : int  +getKg() : int  +setCal() : void  +setCost() : void  +setKg() :void |  |  | | --- | | Melon | | -info : string | | +getInfo() : string  +setInfo() : void  +toString(): string | |

**public** **class** Melon\_Test {

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

Melon m1 = **new** Melon(124, 21, 2, "jjh\_fram");

Melon m2 = **new** Melon(1, 1, 1, "0");

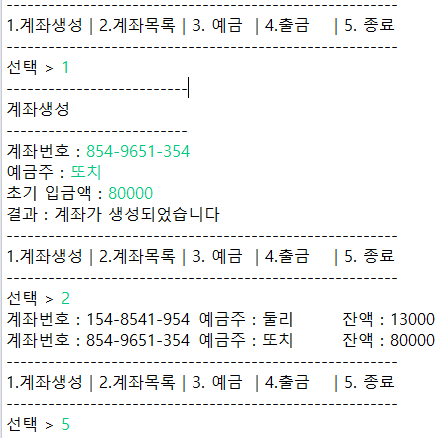
}

}

|  |
| --- |
| **[프로그램 소스]**  class Food {  private int cal;  private int cost;  private int kg;    public Food(int cal, int cost, int kg) {  this.cal = cal;  this.cost = cost;  this.kg = kg;  }    public int getCal() { return cal; }  public int getCost() { return cost; }  public int getKg() { return kg; }  public void setCal(int cal) { this.cal = cal; }  public void setCost(int cost) { this.cost = cost; }  public void setKg(int kg) { this.kg = kg; }  }  class Melon extends Food {  private String info;    public Melon(int cal, int cost, int kg, String info) {  super(cal, cost, kg);  this.info = info;  }    public String getInfo() { return info; }  public void setInfo(String info) { this.info = info; }    public String toString() {  String str = "Melon의 정보";  str += "\n칼로리 : " + this.getCal();  str += "\n가격 : " + this.getCost();  str += "\n중량 : " + this.getKg();  str += "\n정보 : " + info;  return str;  }  }  public class J1\_1122\_HW3 {  public static void main(String[] args) {  Melon m1 = new Melon(124, 21, 2, "jjh\_fram");  Melon m2 = new Melon(100, 210, 21, "우아아앙 구아아악");    System.***out***.println("m1 객체 정보 >>>>");  System.***out***.println(m1);  System.***out***.println();  System.***out***.println("m2 객체 정보 >>>>");  System.***out***.println(m2);  }  } |
| **[실행 결과]** |

1. 키보드로 계좌정보를 입력 받아 계좌를 관리하는 프로그램을 작성하시오. – 교재6장 확인 학습 20번





|  |
| --- |
| **[프로그램 소스]**  import java.util.Scanner;  class Human {  private String bankNum;  private String name;  private int money;  public Human() {  bankNum = "000-0000-000";  name = "NULL";  money = 0;  }    public Human(String bankNum, String name, int money) {  this.bankNum = bankNum;  this.name = name;  this.money = money;  }  public String getName() { return name; }  public String getBankNum() { return bankNum; }  public int getMoney() { return money; }  public void increaseM(int m) { money += m; }  public void decreaseM(int m) { money -= m; }    public String toString() {  String str = "계좌번호 : " + bankNum;  str += "\t예금주 : " + name;  str += "\t잔액 : " + money;  return str;  }  }  class Calculate {  public static void print(String[] menu) {  System.***out***.println("---------------------------------------------");  System.***out***.print("1. " + menu[0]);  for(int i = 1; i < menu.length; i++) {  System.***out***.print(" | ");  System.***out***.print((i+1) + ". " + menu[i]);  }  System.***out***.println("\n---------------------------------------------");  }  }  public class J1\_1122\_HW4 {  public static void main(String[] args) {  Scanner sc = new Scanner(System.***in***);  String[] menu = { "계좌생성", "계좌목록", "예금", "출금", "종료" };  boolean cnd = true;  Human[] h = new Human[10];  String num, name;  int money, i, cnt = 0;  while (cnd) {  Calculate.*print*(menu);  System.***out***.print("선택 > ");  switch (sc.nextInt()) {  case 1:  System.***out***.println("=================");  System.***out***.println(menu[0]);  System.***out***.println("=================");    System.***out***.print("계좌번호 : ");  num = sc.next();  System.***out***.print("예금주 : ");  name = sc.next();  System.***out***.print("초기 입금액 : ");  money = sc.nextInt();  h[cnt] = new Human(num, name, money);    if(h[cnt].getName() != "NULL") {  System.***out***.println("결과 : 계좌가 생성되었습니다.");  cnt++;  }  break;  case 2:  System.***out***.println("=================");  System.***out***.println(menu[1]);  System.***out***.println("=================");    for(i = 0; i < cnt; i++) {  System.***out***.println(h[i]);  }  break;  case 3:  System.***out***.println("=================");  System.***out***.println(menu[2]);  System.***out***.println("=================");    System.***out***.print("계좌번호 : ");  num = sc.next();  System.***out***.print("입금액 : ");  money = sc.nextInt();    for(i = 0; i < cnt; i++) {  if(num.equals(h[i].getBankNum())) {  h[i].increaseM(money);  System.***out***.println("결과 : 입금이 성공되었습니다.");  }  }  break;  case 4:  System.***out***.println("=================");  System.***out***.println(menu[3]);  System.***out***.println("=================");    System.***out***.print("계좌번호 : ");  num = sc.next();  System.***out***.print("출금액 : ");  money = sc.nextInt();    for(i = 0; i < cnt; i++) {  if(num.equals(h[i].getBankNum())) {  h[i].decreaseM(money);  System.***out***.println("결과 : 출금이 성공되었습니다.");  }  }  break;  case 5:  System.***out***.println("Program Exit..");  cnd = false;  continue;  default:  break;  }  }  }  } |
| **[실행 결과]** |