|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **[유헬스2\_3주\_실습과제]** | **학번:** | **20175105** | **이름:** | **곽영주** |

* 정적 변수와 정적 메소드

**public** **class** Employee {

//필드 선언(전용멤버) : 이름(String, name), 급여(int, salary) 객체수(int, count, static) – 본인작성

//생성자 – 매개변수로 받은 값(이름, 급여)을 필드값으로 초기화, 객체수(count)를 1증가 – 본인작성

//객체수(count) 필드에 대한 접근자 메소드(static) – 본인작성

//급여(salary)필드에 대한 접근자 메소드 – 본인 작성

**public** String toString(){

**return** "이름: " + name +"\t급여: " + salary;

}

}

//객체를 크기순으로 정렬

**public** **class** ArraySort {

**public** **static** **void** dim\_sort(***Employee[] dim***){ //매개변수로 객체 배열을 받음

**for**(**int** i=0; i<dim.length-1;i++){

**int** least=i;

**for**(**int** j=i+1;j<dim.length;j++){

//가장 작은 연봉을 갖는 객체를 탐색-본인작성

}

//객체 교환

Employee temp=dim[i];

dim[i]=dim[least];

dim[least]=temp;

}

}

}

//객체 배열을 사용하여 처리

**import** java.util.\*;

**public** **class** EmployeeTest {

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

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

Employee[] obj = **new** Employee[4]; //객체 배열 선언 & 생성

**for**(**int** i=0; i<obj.length;i++){

System.*out*.print("\n이름 입력 : ");

String name=key.nextLine();

System.*out*.print("월급 입력 : ");

**int** sal=key.nextInt();

key.nextLine();

//사용자로부터 입력 받은 값을 생성자 형식매개변수로 전달

//생성된 객체의 참조를 객체 배열 i번째에 저장

obj[i] = **new** Employee(name, sal);

}

//객체 배열 원소 출력

**for**(Employee em : obj)

System.*out*.println(em);

//연봉을 기준으로 정렬한 후 출력 – 본인작성

//필드 count 값 출력 – 본인작성

}

}

|  |
| --- |
| **[프로그램 소스]**  **public** **class** Employee {  **private** String name;  **private** **int** salary;  **private** **static** **int** *count*;    **public** Employee(String name, **int** salary) {  **this**.name = name;  **this**.salary = salary;  *count*++;  }    **public** **static** **int** getCount() {  **return** *count*;  }    **public** **int** getSalary() {  **return** salary;  }    **public** String toString() {  **return** "이름: " + name + "\t급여: " + salary;  }  }  **public** **class** ArraySort {  **public** **static** **void** dim\_sort(Employee[] dim) {  **for**(**int** i=0; i<dim.length-1; i++) {  **int** least = i;  **for**(**int** j=i+1; j<dim.length; j++) {  **if**(dim[least].getSalary() > dim[j].getSalary())  least = j;  }  Employee temp = dim[i];  dim[i] = dim[least];  dim[least] = temp;  }  }  }  **import** java.util.\*;  **public** **class** EmployeeTest {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  Employee[] obj = **new** Employee[4];    **for**(**int** i=0; i<obj.length; i++) {  System.***out***.print("\n이름 입력 : ");  String name = key.nextLine();    System.***out***.print("월급 입력 : ");  **int** sal = key.nextInt();  key.nextLine();    obj[i] = **new** Employee(name, sal);  }    **for**(Employee em : obj)  System.***out***.println(em);    System.***out***.println("\n---연봉을 기준으로 정렬 후 출력---");  ArraySort.*dim\_sort*(obj);  **for**(Employee em : obj)  System.***out***.println(em);    System.***out***.println("\n객체 수: " + Employee.*getCount*());  }  } |
| **[실행 결과]** |

* 상속
* 기존 클래스를 확장해서 새로운 클래스를 만드는 기술
* 다중 상속 불가
* 필드와 메소드 재사용
* 코드 중복을 줄일 수 있음
* 부모클래스
* 자식클래스 : 상속을 받는 클래스
  + 자식 클래스는 부모 클래스의 필드와 메소드 사용 가능
  + ***자식 클래스에서 추가된 멤버 변수나 메소드는 부모 클래스에 의해 사용될 수 없다***
* 상속과 생성자 : 부모 클래스의 생성자가 먼저 실행되고 자식 클래스의 생성자 실행됨
* super 키워드
* 자식클래스에서 부모 클래스의 멤버를 접근할 때 사용되는 부모클래스 타입의 레퍼런스.
* 상속관계에 있는 자식 클래스에서만 사용됨
* 사용 조건
  + 오버라이딩 된 부모 클래스의 메소드 호출
    - 형식 : super.메소드();
  + 자식 클래스에서 동일한 이름의 부모 클래스 필드 참조
    - 형식 :super.필드;
  + 부모 클래스의 생성자 호출
    - 형식 : super();

**class** Account{

**protected** String aname; //예금주

**protected** String anum; //계좌번호

**protected** **int** balance; //잔액

**public** Account(String aname, String anum, **int** balance) {

**this**.aname = aname;

**this**.anum=anum;

**this**.balance=balance;

}

**void** deposit(**int** money) { //입금

balance += money;

}

**int** withdraw(**int** money) { //출금

**if**(money > balance)

**return** 0; //잔고보다 인출액이 많으면 0 반환

**else** {

balance -=money; //인출액 반환

**return** money;

}

}

**public** String toString() {

**return** "예금주 : " + aname +"\t잔액 : " + balance;

}

}

**class** checkAccount **extends** Account{ //Account 클래스를 상속

**private** String cNo; //카드 번호

**public** checkAccount(String name, String num, **int** bal, String cNo) {

**super**(name, num, bal); //수퍼클래스 생성자 호출

**this**.cNo = cNo;

}

**public** String toString() {

//부모 클래스 메소드 호출 시 super 사용

**return** **super**.toString()+"\t카드 번호 : " + cNo;

}

}

**public** **class** InheritanceTest {

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

checkAccount obj=**new** checkAccount("상속", "23-45363-12", 20000, "231-35234-435");

Account superobj = **new** Account("부모클래스", "345-234-2", 50000);

System.***out***.println("계좌 번호 : " + obj.anum); //부모 클래스 필드 접근

System.***out***.print("부모클래스===\n" + superobj);

System.***out***.print("\n자식클래스===\n" + obj);

}

}

|  |
| --- |
| **[실행 결과]** |

* 프로그램 과제 1 : Person 클래스를 상속받는 Customer 클래스를 작성하고 테스트 하시오
* Person 클래스
* 필드: 이름, 주소, 전화번호, private
* 메소드:

-생성자 : 매개변수로 필드 초기화

-접근자, 설정자, toString() 재정의하여 객체 내용 출력

* Customer 클래스
* 필드:고객번호, 마일리지, private
* 메소드 :

-생성자 : 매개변수로 필드 초기화

-접근자, 설정자, toString() 재정의하여 객체 내용 출력

-addMileage(int) : 마일리지 누적, 반환값 없고, 매개변수 값만큼 마일리지를 누적한다.

class Answer1{

public static void main(String[] args){

//Person 객체 생성, 필드 초기값은 생성자 파라메터로 전달

//Customer 객체 생성, 필드 초기값은 생성자 파라메터로 전달

//Customer 객체의 마일리지를 입력된 값만큼 누적

// 객체 내용 출력 – 부모클래스, 자식 클래스

}

}

|  |
| --- |
| **[프로그램 소스]**  **public** **class** Person {  **private** String name;  **private** String address;  **private** String phoneNum;    **public** Person(String name, String address, String phoneNum) {  **this**.name = name;  **this**.address = address;  **this**.phoneNum =phoneNum;  }    **public** String getName() {  **return** name;  }  **public** **void** setName(String name) {  **this**.name = name;  }    **public** String getAddress() {  **return** address;  }  **public** **void** setAddress(String address) {  **this**.address = address;  }    **public** String getPhoneNum() {  **return** phoneNum;  }  **public** **void** setPhoneNum(String phoneNum) {  **this**.phoneNum = phoneNum;  }    **public** String toString() {  **return** "이름: " + name + "\t주소: " + address + "\t전화번호: " + phoneNum;  }  }  **public** **class** Customer **extends** Person {  **private** String customerNum;  **private** **int** mileage;    **public** Customer(String name, String address, String phoneNum, String customerNum, **int** mileage) {  **super**(name,address,phoneNum);  **this**.customerNum = customerNum;  **this**.mileage = mileage;  }  **public** String getCustomerNum() {  **return** customerNum;  }  **public** **void** setCustomerNum(String customerNum) {  **this**.customerNum = customerNum;  }  **public** **int** getMileage() {  **return** mileage;  }  **public** **void** setMileage(**int** mileage) {  **this**.mileage = mileage;  }    **public** **void** addMileage(**int** mileage) {  **this**.mileage += mileage;  }    **public** String toString() {  **return** **super**.toString() + "\n고객번호: " + customerNum + "\t마일리지: " + mileage;  }  }  **import** java.util.\*;  **public** **class** Answer1 {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  Person superobj = **new** Person("곽영주","남양주시 진접읍","010-3329-4331");  Customer obj = **new** Customer("곽영주","남양주시 진접읍","010-3329-4331","98-1",10000);    System.***out***.println("누적 전 마일리지: " + obj.getMileage());    System.***out***.print("추가 마일리지 입력: ");  obj.addMileage(key.nextInt());    System.***out***.println("부모클래스===\n" + superobj);  System.***out***.println("자식클래스===\n" + obj);  }  } |
| **[실행 결과]** |

* 프로그램 과제 2 : 일반적인 휴대폰을 나타내는 Phone 클래스를 작성한다. Phone에는 제조사, 가격, 통신타입(2g 또는 3g) 등의 정보가 저장되어 있다. Phone에서 상속받아서 SmartPhone 클래스를 작성하여 보자. SmartPhone 클래스에는 운영체제 타입, 운영체제 버전, 내부 메모리 크기, 카메라 장착 여부, 블루투스 지원 여부 등의 필드가 추가된다. 생성자, 접근자, 설정자를 포함하여서 각각의 클래스를 작성한다. 이들 클래스들의 객체를 만들고 각 객체의 모든 정보를 출력하는 테스트 클래스를 작성하라.

class Phone{

//필드 선언 – 전용멤버 ,제조사, 가격, 통신타입

//생성자 – 매개변수로 받은 값을 필드로 초기화

//객체 내용을 문자열로 반환하는 toString() 메소드

}

class SmartPhone { //Phone 클래스 상속

//필드 선언 – 전용 멤버, 운영체제 타입, 버전, 내부 메모리크기, 블르투스 지원여부

//생성자 – 매개변수로 받은 값을 필드로 초기화. 단, 부모클래스 필드를 초기화 하기위한 값도 매개변수로 받아서 처리

//객체 내용을 문자열로 반환하는 toString() 메소드. 단, 부모클래스 내용도 포함 할 것

}

class Answer2{

public static void main(String[] args){

//부모 클래스 객체 생성, 필드 초기값은 생성자 매개변수로 전달

//자식 클래스 객체 생성, 필드 초기값은 생성자 매개변수로 전달

//객체 내용 출력 – 부모클래스, 자식 클래스

}

}

|  |
| --- |
| **[프로그램 소스]**  **public** **class** Phone {  **private** String manufacturer;  **private** **int** price;  **private** String comType;    **public** Phone(String manufacturer, **int** price, String comType) {  **this**.manufacturer = manufacturer;  **this**.price = price;  **this**.comType = comType;  }    **public** String getManufacturer() {  **return** manufacturer;  }  **public** **void** setManufacturer(String manufacturer) {  **this**.manufacturer = manufacturer;  }  **public** **int** getPrice() {  **return** price;  }  **public** **void** setPrice(**int** price) {  **this**.price = price;  }  **public** String getComType() {  **return** comType;  }  **public** **void** setConType(String comType) {  **this**.comType = comType;  }  **public** String toString() {  **return** "\n제조사: " + manufacturer + "\t가격: " + price + "만원" + "\t통신타입: " + comType;  }  }  **public** **class** SmartPhone **extends** Phone {  **private** String osType;  **private** **double** version;  **private** **int** memory;  **private** **char** camera;  **private** **char** bluetooth;    **public** SmartPhone(String manufacturer, **int** price, String comType,  String osType, **double** version, **int** memory, **char** camera, **char** bluetooth) {  **super**(manufacturer,price,comType);  **this**.osType = osType;  **this**.version = version;  **this**.memory = memory;  **this**.camera = camera;  **this**.bluetooth = bluetooth;  }  **public** String getOsType() {  **return** osType;  }  **public** **void** setOsType(String osType) {  **this**.osType = osType;  }  **public** **double** getVersion() {  **return** version;  }  **public** **void** setVersion(**double** version) {  **this**.version = version;  }  **public** **int** getMemory() {  **return** memory;  }  **public** **void** setMemory(**int** memory) {  **this**.memory = memory;  }  **public** **char** getCamera() {  **return** camera;  }  **public** **void** setCamera(**char** camera) {  **this**.camera = camera;  }  **public** **char** getBluetooth() {  **return** bluetooth;  }  **public** **void** setBluetooth(**char** bluetooth) {  **this**.bluetooth = bluetooth;  }    **public** String toString() {  String str = **super**.toString();  str += "\n운영체제 타입: " + osType;  str += "\t운영체제 버전: " + version;  str += "\t내부 메모리 크기: " + memory + "GB";  str += "\n카메라 장착 여부: " + camera;  str += "\t블루투스 지원 여부: " + bluetooth;  **return** str;  }  }  **public** **class** Answer2 {  **public** **static** **void** main(String[] args) {  Phone superobj = **new** Phone("Samsung",100,"LTE");  SmartPhone obj = **new** SmartPhone("Samsung",100,"LTE",  "안드로이드",7.0,32,'o','o');    System.***out***.println("부모클래스===" + superobj);  System.***out***.print("\n자식클래스===" + obj);  }  } |
| **[실행 결과]** |

* 프로그램 과제 3 : 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.

|  |
| --- |
| **[프로그램 소스]**  **public** **class** Food {  **private** **double** calorie;  **private** **int** price;  **private** **double** weight;    **public** Food(**double** calorie, **int** price, **double** weight) {  **this**.calorie = calorie;  **this**.price = price;  **this**.weight = weight;  }    **public** **double** getCalorie() {  **return** calorie;  }  **public** **void** setCalorie(**double** calorie) {  **this**.calorie = calorie;  }    **public** **int** getPrice() {  **return** price;  }  **public** **void** setPrice(**int** price) {  **this**.price = price;  }    **public** **double** getWeight() {  **return** weight;  }  **public** **void** setWeight(**double** weight) {  **this**.weight = weight;  }    **public** String toString() {  **return** "칼로리: " + calorie + "\t가격: " + price + "\t중량: " + weight;  }  }  **public** **class** Melon **extends** Food {  **private** String farm;    **public** Melon(**double** calorie, **int** price, **double** weight, String farm){  **super**(calorie, price, weight);  **this**.farm = farm;  }  **public** String getFarm() {  **return** farm;  }  **public** **void** setFarm(String farm) {  **this**.farm = farm;  }    **public** String toString() {  **return** **super**.toString() + "\t경작 농원 정보: " + farm;  }  }  **public** **class** Answer3 {  **public** **static** **void** main(String[] args) {  Food superobj = **new** Food(232.0, 2500, 19.8);  Melon obj = **new** Melon(232.0, 2500, 19.8, "영주농원");    System.***out***.print("부모클래스===\n" + superobj);  System.***out***.print("\n자식클래스===\n" + obj);  }  } |
| **[실행 결과]** |

* 프로그램 과제 4 : 상속 예제 프로그램에서 잔액이 가장 큰 객체를 반환하는 메소드와 매개변수로 전달된 문자열과 일치하는 계좌번호를 갖는 객체를 반환하는 프로그램을 작성하시오.

class SearchObject{

//객체 배열을 매개변수로 받아 잔액이 가장 큰 객체를 반환하는 정적메소드 – 본인작성

//객체 배열과 문자열을 매개변수로 받아 문자열과 일치하는 계좌번호를 갖는 객체를 반환하는 정적메소드 – 본인작성

}

class Answer4{

public static void main(String[] args){

// checkAccount 객체 배열 선언 및 생성, 크기는 5, 초기화는 임의로 할 것

//잔액이 가장 큰 객체 출력

//해당 계좌번화를 입력 받아 메소드로 전달하여 일치하는 객체를 출력

}

}

|  |
| --- |
| **[프로그램 소스]**  **public** **class** SearchObject {    **public** **static** checkAccount large\_bal(checkAccount[] obj) {  **int** large = obj[0].balance;  **int** num = 0;  **for**(**int** i=1; i<obj.length; i++) {  **if**(large < obj[i].balance) {  large = obj[i].balance;  num = i;  }  }  **return** obj[num];  }    **public** **static** checkAccount com\_anum(checkAccount[] obj, String str) {  checkAccount com = **null**;  **for**(**int** i=0; i<obj.length; i++) {  **if**(str.equals(obj[i].anum))  com = obj[i];  }  **return** com;  }  }  **import** java.util.\*;  **public** **class** Answer4 {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  checkAccount[] obj = **new** checkAccount[5];    **for**(**int** i=0; i<obj.length; i++) {  System.***out***.println((i+1) + ". 예금주, 계좌번호, 잔액, 카드번호 순으로 입력: ");  obj[i] = **new** checkAccount(key.next(),key.next(),key.nextInt(),key.next());  }    **for**(checkAccount data : obj)  System.***out***.println(data);  System.***out***.println();    System.***out***.println("---잔액이 가장 큰 객체---");  checkAccount x = SearchObject.*large\_bal*(obj);  System.***out***.println(x);  System.***out***.println();    System.***out***.println("계좌번호 입력: ");  String str = key.next();  checkAccount y = SearchObject.*com\_anum*(obj,str);  System.***out***.println(y);  }  } |
| **[실행 결과]** |