**13주차\_실습과제(학번: 20157135이름:이상우)**

* String 클래스 – 문자열 처리

public class StringTest {

public static void main(String[] args) {

String proverb = "A barking dog"; **// new 연산자 생략**

String s1;

char ch;

boolean b;

int i = Integer.parseInt("123"); **// 문자열 “123”을 숫자 123으로 변환**

double d = Double.parseDouble("3.14"); **// 문자열 "3.14”을 숫자 3.14로 변환**

System.out.println("문자열의 길이 =" + proverb.length());

ch = proverb.charAt(3); **//4번째 문자 반환**

s1 = proverb.toUpperCase(); **// 대문자로 변환**

b= proverb.equals(“a barking Dog”); **//문자열 비교, 같은면 true 다르면 false 반환**

System.out.println(s1);

System.out.println(ch);

System.out.println(b);

System.out.println(“결과 : “ + (i+d));

}

}

|  |
| --- |
| **[실행결과]** |

* 생성자, 설정자, 접근자, 접근제어

- 접근제어 : public(공용멤버) , private(전용멤버) , protected, 생략(package 멤버)

**class** Student{

//접근제어가 private 이면 클래스 내부에서만 사용할 수 있다.

**private** String name; //이름을 저장하기 위한 필드 선언, 전용 멤버

**private** **int** java, eng; //과목을 저장하기 위한 필드 선언, 전용 멤버

//디폴트 생성자 - 매개변수 없음

//생성자는 반드시 클래스명과 일치 해야 한다

//생성자는 객체 생성 시 딱 한번만 호출

**public** Student(){

**this**("", 0, 0); //클래스내에 오버로딩된 생성자 호출, 반드시 첫 문장에 작성되야 함

}

//객체를 초기화 하는 생성자

**public** Student(String name, **int** java, **int** eng){

**this**.name = name; //필드와 매개 변수명이 같으면 필드명 앞에 this를 사용

**this**.java=java;

**this**.eng=eng;

}

**public** String getName() { //접근자 메소드 : 필드 반환

**return** name;

}

**public** **void** setName(String name) { //설정자 메소드 : 필드 초기화 & 변경

**this**.name = name;

}

**public** **int** getEng() {

**return** eng;

}

**public** **void** setEng(**int** eng) {

**this**.eng = eng;

}

**public** **int** hap(){ //과목 합을 계산하여 반환하는 메소드

**return** java+eng;

}

**public** String toString(){ //객체 내용(이름, 과목성적, 합)을 문자열로 반환하는 메소드

String str="이름 : " + name;

str += "\njava : " + java + "\teng : " + eng;

str += "\n과목합 : " + hap(); //클래스 내의 메소드 호출

**return** str;

}

}

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

**public** **class** StudentTest {

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

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

Student std;

//Student 클래스 객체 obj 선언 & 생성

Student obj=**new** Student("김낙준", 90, 80);

//표준 입력장치로 입력된 값을 필드값으로 초기화

System.***out***.println("이름과 두 과목 성적을 입력하세요 :");

String name=in.next();

**int** java=in.nextInt();

**int** eng=in.nextInt();

//객체 생성시 생성자 매개변수로 입력 받은 값을 전달

std=**new** Student(name, java, eng);

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

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

}

}

|  |
| --- |
| **[실행결과]** |

* **중간 점검**

1. 다음 프로그램에서 잘못된 부분을 모두 지적하고 올바르게 수정하시오. 그리고 수정된 후의 출력 결과를 쓰시오.

**class** Television {

**private** String model;

**void** setModel(String b) { // 설정자

model = b;

}

**void** getModel() { // 접근자

**return** model;

}

}

**public** **class** TelevisionTest {

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

Television t = **new** Television;

t.setModel("STV-101");

String b = getModel();

}

}

1. 접근자에는 void 를 쓰면 안된다.
2. 생성자 마지막에 ()를 해줘야 한다.
3. 3. 접근자에 return 값의 자료형을 써줘야 한다, 문제의 자료형은 String
4. 4. getModel()메소드 호출시에는 다른 클레스에 있음으로 객체참조변수를 통해 접근한다.

변수는 t 임으로 t. getModel() 로 호출 한다.

1. 출력문이 없다.

|  |
| --- |
| **[수정된 소스]**  **class** Television {  **private** String model;  **void** setModel(String b) { // 설정자  model = b;  }  String getModel() { // 접근자  **return** model;  }  }  **public** **class** t03 {  **public** **static** **void** main(String[] args){  Television t = **new** Television();  t.setModel("STV-101");  String b = t.getModel();  }  } |
| **[실행 결과]** |

1. 은행 계좌를 처리하는 부분 클래스이다. 완성하시오.

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

String name; **//이름을 저장하기 위한 필드 name 선언, 전용멤버**

int balance; **//잔액을 저장하기 위한 필드 balance 선언 전용멤버**

**//생성자 : 매개변수로 받은 값을 필드(이름, 잔액)로 초기화 – 본인작성**

**//필드에 대한 설정자 메소드 – 본인작성**

**//필드에 대한 접근자 메소드 – 본인작성**

**//매개변수로 받은 값만큼 잔액을 증가하는 메소드, 반환값 없음**

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

balance += money;

}

**//매개변수로 받은 값만큼 잔액을 감소하는 메소드, 반환값 없음**

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

balace += money

}

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

**return** "이름 : " + name + "\t잔액 : " + balance;

}

}

1. 2번의 클래스를 테스트 하는 AccountTest 클래스를 작성하시오

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

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

**//Account 클래스 객체 obj 선언**

**//표준입력 장치로 입력 받은 값을 Account 객체 생성시, 생성자 매개변수로 전달 – 본인작성**

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

**//dePosit() 메소드를 호출하여 잔액을 증가 – 본인작성**

**//접근자 메소드를 사용하여 이름과 잔액을 출력하는 문장 – 본인작성**

}

}

|  |
| --- |
| **[2번과3번 - 프로그램 소스]**  **import** java.util.\*;  **class** Account {  **private** String name; //이름을 저장하기 위한 필드 name 선언, 전용멤버  **private** **int** balance; //잔액을 저장하기 위한 필드 balance 선언 전용멤버    //생성자 : 매개변수로 받은 값을 필드(이름, 잔액)로 초기화 – 본인작성  **public** Account(String name , **int** balance){  **this**.name = name;  **this**.balance = balance;  }  //필드에 대한 설정자 메소드 – 본인작성  **public** **void** namebal(){    }  //필드에 대한 접근자 메소드 – 본인작성  **public** String name(){  **return** name;  }  **public** **int** balance(){  **return** balance;  }  //매개변수로 받은 값만큼 잔액을 증가하는 메소드, 반환값 없음  **void** dePosit(**int** money){  balance += money;  }  //매개변수로 받은 값만큼 잔액을 감소하는 메소드, 반환값 없음  **void** withDraw(**int** money){  balance += money;  }    **public** String toString(){ //객체 내용(이름과 잔액)을 문자열로 반환  **return** "이름 : " + name + "\t잔액 : " + balance;  }  }  **public** **class** t04 {  **public** **static** **void** main(String[] args) {  Scanner key =**new** Scanner(System.***in***);  Account obj;  //Account 클래스 객체 obj 선언  System.***out***.print("이름,잔액 입력");  obj = **new** Account(key.next(), key.nextInt());  //표준입력 장치로 입력 받은 값을 Account 객체 생성시, 생성자 매개변수로 전달 – 본인작성    System.***out***.println(obj); //객체 내용 출력  //dePosit() 메소드를 호출하여 잔액을 증가 – 본인작성  //접근자 메소드를 사용하여 이름과 잔액을 출력하는 문장 – 본인작성  }  } |
| **[실행 결과]** |

1. 제시된 프로그램을 조건대로 수정하고 실행하여 결과를 확인하시오

* Rectangle 클래스
* 전용 필드로 수정하시오
* 매개변수를 필드값으로 초기화하는 생성자를 추가하시오
* 필드에 대한 설정자와 접근자 메소드를 추가하시오
* RectangleTest 클래스
* 입력 받은 값을 생성자 매개변수로 전달하도록 수정

|  |
| --- |
| //사각형 넓이를 계산하기 위한 클래스  **public** **class** Rectangle {  **int** ga; //가로  **int** se; //세로    **int** getArea() { //넓이를 계산하여 반환  **int** result=ga\*se;  **return** result;  }    **public** String toString(){ //객체 내용을 문자열로 반환  String str="가로 : " + ga;  str +="\n세로 : " + se;  str +="\n넓이 : " + getArea(); //넓이를 계산하여 반환하는 메소드호출  **return** str;  }  }  //Rectangle 클래스의 객체를 생성하고 테스트하는 클래스  **public** **class** RectangleTest {  **public** **static** **void** main(String[] args) {  Rectangle obj=**new** Rectangle();  Scanner key=**new** Scanner(System.*in*);  System.*out*.println("가로를 입력하세요");  obj.ga=key.nextInt(); //입력 받은 값을 ga 필드로 초기화  System.*out*.println("세로를 입력하세요");  obj.se=key.nextInt();//입력 받은 값을 se 필드로 초기화  System.*out*.println(obj); //객체 내용 출력  }  } |
| **[수정된 프로그램 소스]**  **import** java.util.\*;  **class** Rectangle {  **private** **int** ga; //가로  **private** **int** se; //세로  **public** Rectangle(**int** ga , **int** se){  **this**.ga=ga;  **this**.se = se;  }  **int** getArea() { //넓이를 계산하여 반환  **int** result=ga\*se;  **return** result;  }  **public** **int** ga(){  **return** ga;  }  **public** **int** se(){  **return** se;  }      **public** String toString(){ //객체 내용을 문자열로 반환  String str="가로 : " + ga;  str +="\n세로 : " + se;  str +="\n넓이 : " + getArea(); //넓이를 계산하여 반환하는 메소드호출  **return** str;  }  }  **public** **class** t05 {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner (System.***in***);  Rectangle obj;  System.***out***.print("가로 세로 입력");  obj=**new** Rectangle(key.nextInt() , key.nextInt() );  System.***out***.println(obj); //객체 내용 출력  }  } |
| **[실행결과]** |

* 프로그램 과제

1. 입력 받은 문자열의 길이를 출력하는 프로그램을 작성하시오. 단, “quit” 가 입력되면 반복을 종료한다. 또한 입력 받은 전체 문자열 개수를 함께 출력

class StringTest1{

public static void main(String[] args){

//문자열 변수 선언

while(true){

//문자열 입력

//입력 받은 문자열이 “quit” 와 같으면 반복문 종료

//문자열 길이 출력

}

}

|  |
| --- |
| **[프로그램 소스]**  **import** java.util.\*;  **public** **class** t06 {  **public** **static** **void** main(String[] args){  Scanner key = **new** Scanner(System.***in***);  String str; //문자열 변수 선언  **while**(**true**){  System.***out***.println("문자열 입력");  str=key.next(); //문자열 입력  **if**(str.equalsIgnoreCase("quit")){  **break**;  //입력 받은 문자열이 “quit” 와 같으면 반복문 종료  }  System.***out***.println("문자열 길이는:"+str.length());  //문자열 길이 출력  }  }  } |
| **[실행결과]** |

1. 다음과 같이 초기화된 문자열에서 표준 입력 장치로 입력된 문자가 몇 번째 있는지 위치 값을 출력하는 프로그램을 작성하시오

String str= “자바 프로그램은 재미 없습니다. 어렵습니다”

[출력예시]

문자열: 자바 프로그램은 재미 없습니다. 어렵습니다

입력문자: 프

위치 : 4

class StringTest2{

public static void main(String[] args){

String str= “자바 프로그램은 재미 없습니다. 어렵습니다”

//문자 변수 선언

//문자 입력

//문자열 길이 계산

//문자열 길이만큼 반복문을 실행하며 입력된 문자와 같은 문자가 있는지 체크 – 힌트

if(입력된 문자 == str.charAt(i))

}

}

|  |
| --- |
| **[프로그램 소스]**  **import** java.util.\*;  **public** **class** t07 {  **public** **static** **void** main(String[] args){  Scanner key = **new** Scanner(System.***in***);  String str = "자바 프로그램은재미 없습니다.어렵습니다";  System.***out***.println("문자입력");  **char** st = key.next().charAt(0);  **int** z=0;  **int** a = str.length();  **for**(**int** i =0;i<a;i++){  **if**(st == str.charAt(i) ){  z=i;  **break**;  }  }  System.***out***.println(str);  System.***out***.println(st);  System.***out***.println(z);  }  } |
| **[실행결과]** |

1. 표준 입력장치로 입력 받은 문자열을 역순으로 출력하는 프로그램을 작성하시오. 단, “finish”가 입력되면 반복을 종료한다

class StringTest2{

public static void main(String[] args){

//문자열 변수 선언

do{

//문자열 입력

//입력된 문자열이 “finish”이면 반복문 종료

//문자열 길이 계산

//for반복문을 사용하여 마지막 문자부터 출력

} while(true);

}

}

|  |
| --- |
| **[프로그램 소스]**  **import** java.util.\*;  **public** **class** t08 {  **public** **static** **void** main(String[] args) {  Scanner key = **new** Scanner(System.***in***);  String str;  **do** {  System.***out***.println("문자열 입력");  str = key.next();  **int** a = str.length();  **if** (str.equalsIgnoreCase("finish"))  **break**;  **for** (**int** j = a - 1; j >= 0; j--) {  System.***out***.print(str.charAt(j));  }  System.***out***.println();  } **while** (**true**);  }  } |
| **[실행결과]** |

1. 책을 나타내는 Book 클래스를 정의하시오 Book 클래스는 제목(title)과 저자(author)를 나타내는 필드를 가진다. 필드는 모두 private로 선언한다. 각 필드에 대하여 접근자와 설정자 메소드를 작성하시오
2. Book 클래스를 테스트하는 BookTest 클래스를 작성하시오. main()에서 설정자와 접근자 메소드를 이용하여 제목과 저자를 설정하고 필드값을 출력하도록 한다.

|  |
| --- |
| **[4번과5번 - 프로그램 소스]**  **class** book{  **private** String name;  **private** String author;  **public** **void** set(String name , String author){  **this**.name = name;  **this**.author=author;  }  **public** String getname(){  **return** name;  }  **public** String getauthor(){  **return** author;  }  }  **public** **class** t09 {  **public** **static** **void** main(String[] args){  book obj = **new** book();  obj.set("홍길동전" , "길동이");  System.***out***.print("제목:"+obj.getname());  System.***out***.print("저자:"+obj.getauthor());  }  } |
| **[실행 결과]** |

1. 애완견을 관리하는 PetRecord 클래스를 작성하시오.

* 필드구성 :이름(String), 나이(int), 몸무게(double) – 모두 전용 필드로 할 것
* 각 필드에 대한 setter와 getter 메소드
* 단, 나이와 몸무게에 대한 setter 메소드에서는 매개변수 값이 0보다 작으면 필드값을 0으로 초기화한다.
* **객체 상태(이름, 나이, 몸무게)를 문자열로 반환하는** toString() 메소드 추가
* 매개변수로 필드를 초기화하는 생성자

1. PetRecord 클래스를 객체를 생성하고 테스트 하시오. 이름과 나이, 몸무게를 입력 받아 객체 생성 시 생성자 매개변수로 전달하여 필드 값을 초기화하도록 한다.

|  |
| --- |
| **[6번과7번 - 프로그램 소스]**  **import** java.util.\*;  **class** PetRecord {  **private** String name;  **private** **int** age;  **private** **double** kg;  **public** **void** setter (String name ,**int** age , **double** kg){  **this**.name = name;  **if**(age < 0)  **this**.age=0;  **this**.age=age;  **if**(kg <0)  **this**.kg=0;  **this**.kg=kg;  }  **public** **int** getage(){  **return** age;  }  **public** **double** getkg(){  **return** kg;  }  **public** String getneme(){  **return** name;  }  **public** String toString(){  **return** "이름:"+getneme() +"나이:"+getage()+"몸무게:"+getkg();  }  }  **public** **class** t10 {  **public** **static** **void** main(String[] args){  Scanner key = **new** Scanner(System.***in***);  PetRecord obj = **new** PetRecord();  System.***out***.println("이름 , 나이 ,무게 입력");  obj.setter(key.next() , key.nextInt(), key.nextDouble());  System.***out***.print(obj.toString());  }  } |
| **[실행 결과]** |