과목 : 객체지향 프로그래밍(가반)

교수 : 최지웅 교수

이름 : 김병준

학번 : 20162448

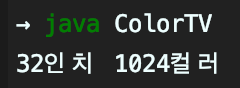
과제 #4

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 12 | 13 | 14 |
| O | O | O | O | O | O | O | O | O | O | O | O |

1. ColorTV.java

|  |
| --- |
| class TV {  private int size;  TV(int size) { this.size = size; }  protected int getSize() { return size; }  }  public class ColorTV extends TV{  private int color;  private ColorTV(int size, int color) {  super(size);  this.color = color;  }  private void printProperty() {  System.out.println(super.getSize() + "인치 " + color + "컬러");  }  public static void main(String[] args) {  ColorTV myTV = new ColorTV(32, 1024);  myTV.printProperty();  }  } |

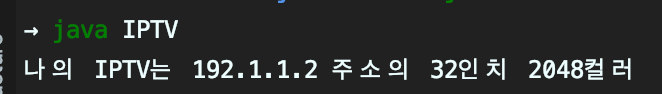
* Result



1. IPTV.java

|  |
| --- |
| class TV {  private int size;  TV(int size) { this.size = size; }  protected int getSize() { return size; }  }  class ColorTV extends TV{  private int color;  ColorTV(int size, int color) {  super(size);  this.color = color;  }  private void printProperty() {  System.out.println(super.getSize() + "인치 " + color + "컬러");  }  public static void main(String[] args) {  ColorTV myTV = new ColorTV(32, 1024);  myTV.printProperty();  }  public int getColor() { return color; }  }  public class IPTV extends ColorTV{  String ip;  private IPTV(String ip, int size, int color) {  super(size, color);  this.ip = ip;  }  public static void main(String[] args) {  IPTV iptv = new IPTV("192.1.1.2", 32, 2048);  iptv.printProperty();  }  private void printProperty() {  System.out.println("나의 IPTV는 " + ip + " 주소의 " + super.getSize() + "인치 " + super.getColor() + "컬러" );  }  } |

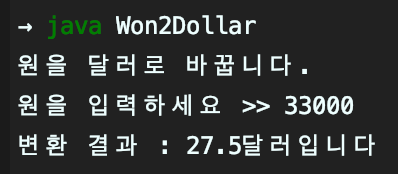
* Result



1. Won2Dollar.java – 원화를 달러로 변환

|  |
| --- |
| import java.util.Scanner;  abstract class Converter {  abstract protected double convert(double src);  abstract protected String getSrcString();  abstract protected String getDestString();  protected double ratio;  public void run() {  Scanner scanner = new Scanner(System.in);  System.out.println(getSrcString() + "을 " + getDestString() + "로 바꿉니다.");  System.out.print(getSrcString() + "을 입력하세요 >> ");  Double val = scanner.nextDouble();  Double res = convert(val);  System.out.println("변환 결과 : " + res + getDestString() + "입니다");  scanner.close();  }  }  public class Won2Dollar extends Converter{  String src = "원";  String dest = "달러";  public Won2Dollar(double ratio) {  this.ratio = ratio;  }  public static void main(String[] args) {  Won2Dollar toDollar = new Won2Dollar(1200);  toDollar.run();  }  protected double convert(double src) {  return src / ratio;  }  protected String getSrcString() {  return src;  }  protected String getDestString() {  return dest;  }  } |

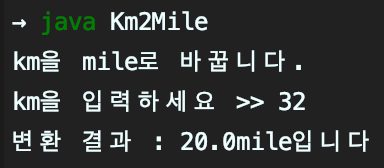
* Result



1. Km2Mile.java – 킬로미터를 마일로 변환

|  |
| --- |
| import java.util.Scanner;  abstract class Converter {  abstract protected double convert(double src);  abstract protected String getSrcString();  abstract protected String getDestString();  protected double ratio;  public void run() {  Scanner scanner = new Scanner(System.in);  System.out.println(getSrcString() + "을 " + getDestString() + "로 바꿉니다.");  System.out.print(getSrcString() + "을 입력하세요 >> ");  Double val = scanner.nextDouble();  Double res = convert(val);  System.out.println("변환 결과 : " + res + getDestString() + "입니다");  scanner.close();  }  }  public class Km2Mile extends Converter {  String src = "km";  String dest = "mile";  private Km2Mile(double ratio) {  this.ratio = ratio;  }  public static void main(String args[]) {  Km2Mile tomile = new Km2Mile(1.6);  tomile.run();  }  protected double convert(double src) {  return src / ratio;  }  protected String getSrcString() {  return src;  }  protected String getDestString() {  return dest;  }  } |

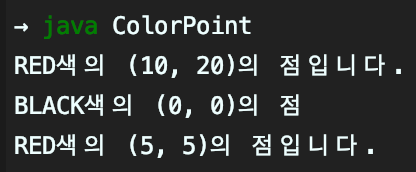
* Result



5~6. ColorPoint.java

|  |
| --- |
| class Point {  private int x, y;  public Point(int x, int y) {  this.x = x;  this.y = y;  }  public int getX() {  return x;  }  public int getY() {  return y;  }  protected void move(int x, int y) {  this.x = x;  this.y = y;  }  }  public class ColorPoint extends Point{  private String color = "BLACK";  public static void main(String[] args) {  // 5번  ColorPoint cp1 = new ColorPoint(5, 5, "YELLOW");  cp1.setXY(10, 20);  cp1.setColor("RED");  String str = cp1.toString();  System.out.println(str + "입니다.");  // 6번  ColorPoint zeroPoint = new ColorPoint();  System.out.println(zeroPoint.toString());  ColorPoint cp2 = new ColorPoint(5, 5);  cp2.setXY(5, 5);  cp2.setColor("RED");  System.out.println(cp2.toString() + "입니다.");  }  private ColorPoint() {  super(0, 0);  }  private ColorPoint(int x, int y) {  super(x, y);  }  private ColorPoint(int x, int y, String color) {  super(x, y);  this.color = color;  }  private void setXY(int x, int y) {  move(x, y);  }  public String toString() {  String string = "";  string += color + "색의 (" + getX() + ", " + getY() + ")의 점";  return string;  }  private void setColor(String color) {  this.color = color;  }  } |

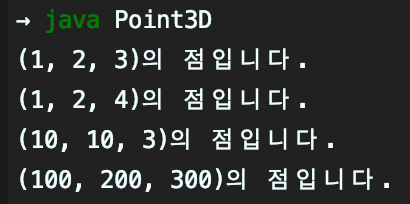
* Result(Line(1) = 5번, Line(2~3) = 6번)



1. Point3D.java

|  |
| --- |
| class Point {  private int x, y;  public Point(int x, int y) {  this.x = x;  this.y = y;  }  public int getX() {  return x;  }  public int getY() {  return y;  }  protected void move(int x, int y) {  this.x = x;  this.y = y;  }  }  public class Point3D extends Point {  private int z;  private Point3D(int x, int y, int z) {  super(x, y);  this.z = z;  }  public static void main(String[] args) {  Point3D p = new Point3D(1, 2, 3);  System.out.println(p.toString() + "입니다.");  p.moveUp();  System.out.println(p.toString() + "입니다.");  p.moveDown();  p.move(10, 10);  System.out.println(p.toString() + "입니다.");  p.move(100, 200, 300);  System.out.println(p.toString() + "입니다.");  }  public String toString() {  String string = "";  string += "(" + getX() + ", " + getY() +", " + getZ() + ")의 점";  return string;  }  private void move(int x, int y, int z) {  move(x, y);  this.z = z;  }  private void moveUp() {  this.z++;  }  private void moveDown() {  this.z--;  }  private int getZ() {  return z;  }  } |

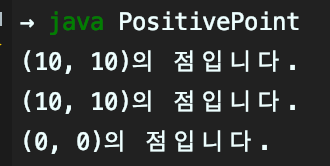
* Result



1. PositivePoint.java

|  |
| --- |
| class Point {  private int x, y;  public Point(int x, int y) {  this.x = x;  this.y = y;  }  public int getX() {  return x;  }  public int getY() {  return y;  }  protected void move(int x, int y) {  this.x = x;  this.y = y;  }  }  public class PositivePoint extends Point{  private PositivePoint() {  super(0, 0);  }  private PositivePoint(int x, int y) {  super(0, 0);  if(x < 0 || y < 0)  move(x, y);  }  public void move(int x, int y) {  if(x >= 0 && y >= 0)  super.move(x, y);  }  public String toString() {  String string = "";  string += "(" + getX() + ", " + getY() + ")의 점";  return string;  }  public static void main(String[] args) {  PositivePoint p = new PositivePoint();  p.move(10, 10);  System.out.println(p.toString() + "입니다.");  p.move(-5, 5);  System.out.println(p.toString() + "입니다.");  PositivePoint p2 = new PositivePoint(-10, -10);  System.out.println(p2.toString() + "입니다.");  }  } |

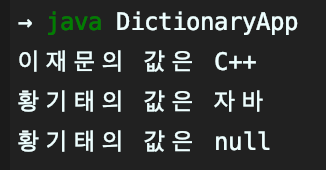
* Result



1. DictionaryApp.java

|  |
| --- |
| abstract class PairMap {  protected String[] keyArray;  protected String[] valueArray;  abstract String get(String key);  abstract void put(String key, String value);  abstract String delete(String key);  abstract int length();  }  class Dictionary extends PairMap {  private int length = 0;  Dictionary(int capacity) {  keyArray = new String[capacity];  valueArray = new String[capacity];  }  @Override  String get(String key) {  for(int i = 0; i < length; i++) {  if(keyArray[i].equals(key)) {  return valueArray[i];  }  }  return null;  }  @Override  void put(String key, String value) {  boolean check = false;  for(int i = 0; i < length; i++) {  if (keyArray[i].equals(key)) {  valueArray[i] = value;  check = true;  break;  }  }  if(!check) {  keyArray[length] = key;  valueArray[length] = value;  length++;  }  }  @Override  String delete(String key) {  for(int i = 0; i < length; i++) {  if (keyArray[i].equals(key)) {  String deletevalue = keyArray[i];  suit(i);  length--;  return deletevalue;  }  }  return null;  }  @Override  int length() {  return length;  }  private void suit(int index) {  for(int i = index; i < this.length; i++) {  keyArray[i] = keyArray[i + 1];  valueArray[i] = valueArray[i + 1];  }  valueArray[length] = "";  }  }  public class DictionaryApp {  public static void main(String[] args) {  Dictionary dic = new Dictionary(10);  dic.put("황기태", "자바");  dic.put("이재문", "파이썬");  dic.put("이재문", "C++");  System.out.println("이재문의 값은 " + dic.get("이재문"));  System.out.println("황기태의 값은 " + dic.get("황기태"));  dic.delete("황기태");  System.out.println("황기태의 값은 " + dic.get("황기태"));  }  } |

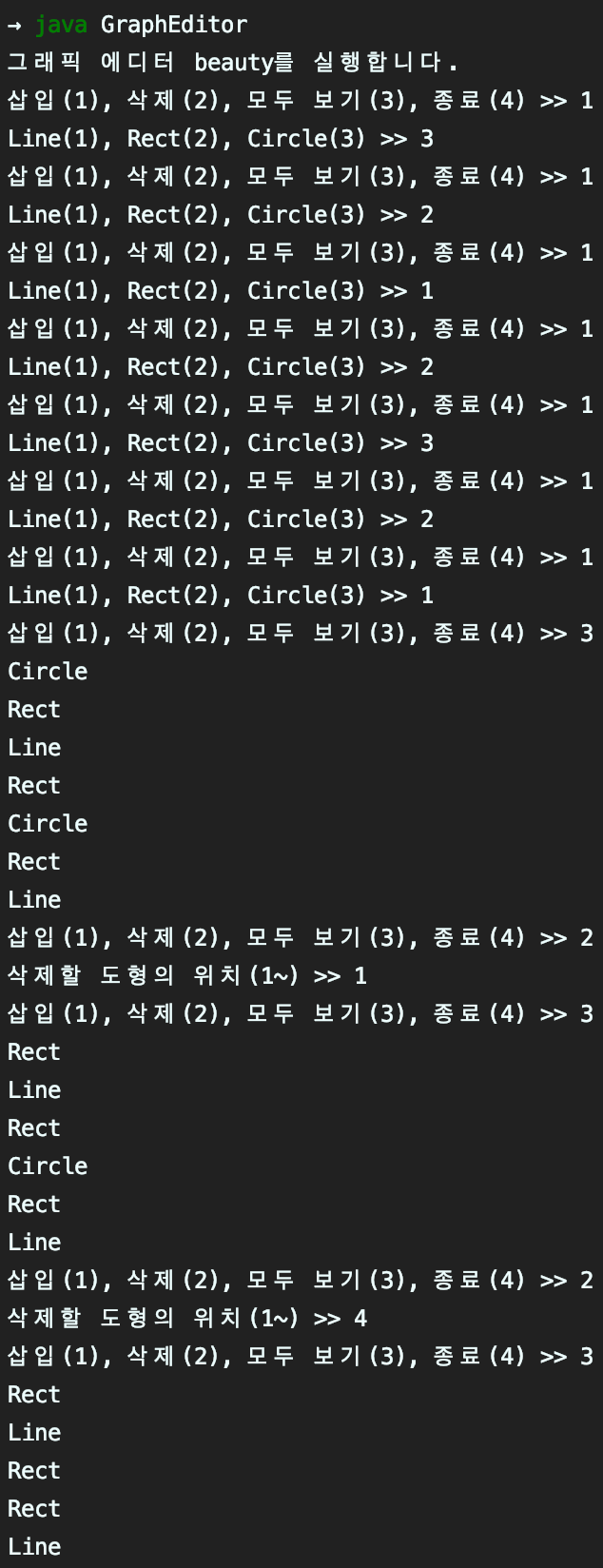
* Result



1. GraphicEditor.java

|  |
| --- |
| import java.util.Scanner;  abstract class Shape {  private Shape next;  public Shape() {next = null;}  public void setNext(Shape obj) {next = obj;}  public Shape getNext() { return next;}  public abstract void draw();  }  class Line extends Shape {  Line() {  super();  }  @Override  public void draw() {  System.out.println("Line");  }  }  class Circle extends Shape {  @Override  public void draw() {  System.out.println("Circle");  }  }  class Rect extends Shape {  @Override  public void draw() {  System.out.println("Rect");  }  }  public class GraphEditor {  private Shape start;  private GraphEditor() {  this.start = null;  }  public static void main(String[] args) {  int input;  GraphEditor graphEditor = new GraphEditor();  Scanner scanner = new Scanner(System.in);  System.out.println("그래픽 에디터 beauty를 실행합니다.");  while (true) {  int select;  System.out.print("삽입(1), 삭제(2), 모두 보기(3), 종료(4) >> ");  input = scanner.nextInt();  switch (input) {  case 1:  System.out.print("Line(1), Rect(2), Circle(3) >> ");  select = scanner.nextInt();  graphEditor.Insert(select);  break;  case 2:  System.out.print("삭제할 도형의 위치(1~) >> ");  select = scanner.nextInt();  if(select <= 0) {  System.out.println("삭제할 수 없습니다.");  break;  }  graphEditor.Delete(select);  break;  case 3:  graphEditor.Print();  break;  case 4:  System.out.println("beauty를 종료합니다.");  System.exit(-1);  break;  }  }  }  private void Insert(int select) {  Shape temp = start;  switch (select) {  case 1:  if(start != null) {  while (temp.getNext() != null) {  temp = temp.getNext();  }  temp.setNext(new Line());  } else  start = new Line();  break;  case 2:  if(start != null) {  while (temp.getNext() != null) {  temp = temp.getNext();  }  temp.setNext(new Rect());  } else  start = new Rect();  break;  case 3:  if(start != null) {  while (temp.getNext() != null) {  temp = temp.getNext();  }  temp.setNext(new Circle());  } else  start = new Circle();  break;  }  }  private void Delete(int select) {  Shape temp = start;  if(Check(select)) {  if(select == 1) {  start = temp.getNext();  } else {  for (int i = 1; i < select - 1; i++) {  temp = temp.getNext();  }  temp.setNext(temp.getNext().getNext());  }  }  }  private boolean Check(int select) {  Shape temp = start;  for(int i = 1; i < select; i++) {  temp = temp.getNext();  }  if(temp == null) {  System.out.println("삭제할 수 없습니다.");  return false;  }  return true;  }  private void Print() {  Shape temp = start;  if(temp == null) {  System.out.println("삽입한 도형이 없습니다");  } else {  while (temp != null) {  temp.draw();  temp = temp.getNext();  }  }  }  } |

* Result



개체, 손목시계이(가) 표시된 사진

자동 생성된 설명

13~14. Figure.java

|  |
| --- |
| interface Shape {  final double PI = 3.14;  void draw();  double getArea();  default public void redraw() {  System.out.print("--- 다시 그립니다. ");  draw();  }  }  class Circle implements Shape {  private double radius;  Circle(double radius) {  this.radius = radius;  }  @Override  public void draw() {  System.out.println("반지름이 " + radius + "인 원입니다.");  }  @Override  public double getArea() {  return radius \* radius \* PI;  }  }  class Oval implements Shape {  double x, y;  Oval(double x, double y) {  this.x = x;  this.y = y;  }  @Override  public void draw() {  System.out.println(x + "\*" + y + "에 내접하는 원입니다.");  }  @Override  public double getArea() {  return x \* y \* PI;  }  }  class Rect implements Shape {  double x, y;  Rect(double x, double y) {  this.x = x;  this.y = y;  }  @Override  public void draw() {  System.out.println(x + "\*" + y + "크기의 사각형입니다.");  }  @Override  public double getArea() {  return x \* y;  }  }  public class Figure {  public static void main(String[] args) {  // 13번  Shape donut = new Circle(10);  donut.redraw();  System.out.println("면적은 " + donut.getArea());  //14번  Shape[] list = new Shape[3];  list[0] = new Circle(10);  list[1] = new Oval(20, 30);  list[2] = new Rect(10, 40);  for(int i = 0; i < list.length; i++)  list[i].redraw();  for(int i = 0; i < list.length; i++)  System.out.println("면적은 " + list[i].getArea());  }  } |

* Result

