단국대학교 자바프로그래밍2   
lab05 보고서

2020-10-19



응용컴퓨터공학과 32160462 김도영

내용

[1. 결과 사진 및 결과 텍스트 2](#_Toc53959768)

[2. 구현 방법 4](#_Toc53959769)

[2.1 목적 4](#_Toc53959770)

[2.2 구현 설계 4](#_Toc53959771)

[2.3 구현 결과 14](#_Toc53959772)

## 결과 사진 및 결과 텍스트

lab05 MAINTEST

please enter select 1 (shape2DCalculator) or 2 (shape3DCalculator)

1

Please enter Triangle Base:10

Please enter Triangle height:20

[calculate Triangle ]: 100.0

Please enter Kite horizontalDiagonal:30

Please enter Kite verticalDiagonal:40

[calculate Kite ]: 600.0

Please enter Paralleogram Base:50

Please enter Paralleogram height:60

[calculate Paralleogram ]: 3000.0

Please enter Rectangle Base:50

Please enter Rectangle height:40

[calculate Rectangle ]: 1000.0

Please enter Square side:30

[calculate Square ]: 900.0

Please enter Trapezoid top:20

Please enter Trapezoid bottom:10

Please enter Trapezoid height:50

[calculate Trapezoid ]: 20000.0

lab05 MAINTEST

please enter select 1 (shape2DCalculator) or 2 (shape3DCalculator)

2

Please enter Cuboid width:10

Please enter Cuboid height:11

Please enter Cuboid length:12

[calculate Cuboid ]: 1210.0

Please enter Cylinder radius:15

Please enter Cylinder height:17

[calculate Cylinder ]: 12016.591899980958

Please enter Sphere radius:50

[calculate Sphere ]: 523598.7755982988

Please enter Pyramid base:10

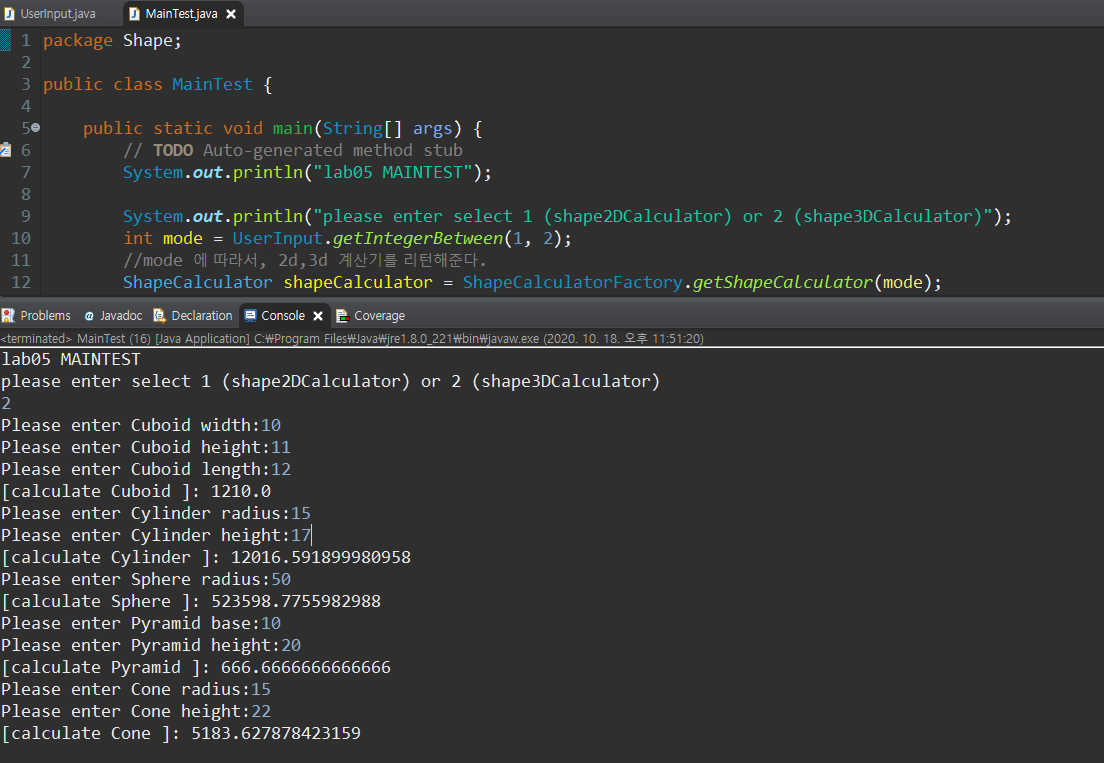
Please enter Pyramid height:20

[calculate Pyramid ]: 666.6666666666666

Please enter Cone radius:15

Please enter Cone height:22

[calculate Cone ]: 5183.627878423159



## 구현 방법

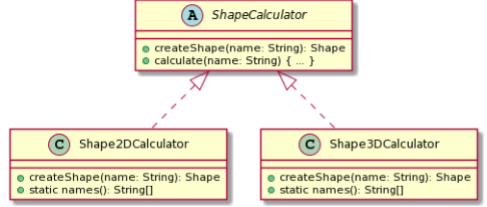
### 2.1 목적

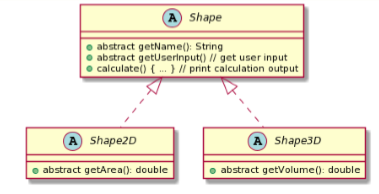
팩토리 패턴, 관련되어 있는 패밀리 오브젝트들을 생성해주는 팩토리 클래스를 만들어서 사용하자.

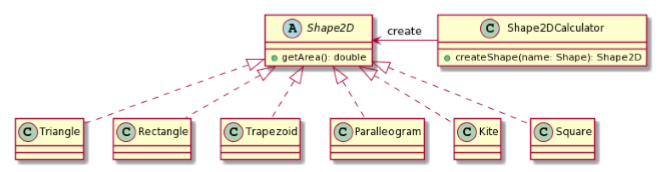
Shape 라는 2d,3d 도형을 생성하는 팩토리 클래스를 만들어 보고 shape팩토리가 가지고 있는 타입별로 생성후에 사용자의 입력을 받아 도형의 값들을 셋팅하여 너비 및 부피를 구한다.

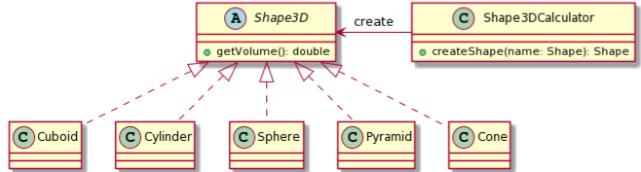
팩토리 팩턴이 적용되는 부분은 도형의 이름에 따라서 다르게 도형객체를 생성해 주는 부분이다.

### 2.2 구현 설계

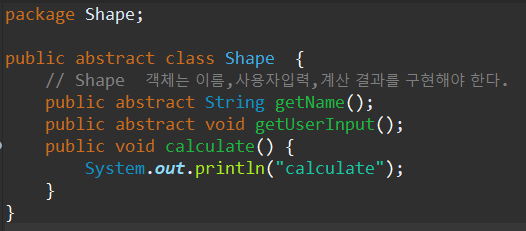
.

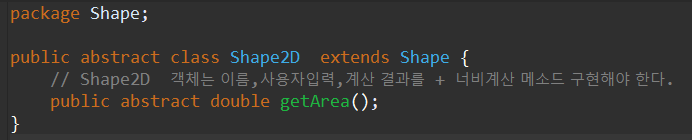


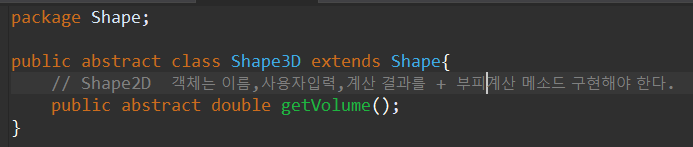




#### 2.2.1 Shape 및 Shape2D Shape3D 구현하기

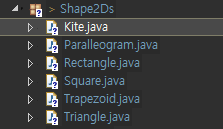






#### 2.2.2 shape2D를 상속받아 concrete class 구현하기

콘크리트 class는 따로 package로 분리했습니다.



2.2.2.1 연꼴형 Shape2D

package Shape.Shape2Ds;

import Shape.Shape2D;

import Shape.UserInput;

*/\**

*\* 연꼴형 클래스*

*\* Kite  horizontalDiagonal,verticalDiagonal*

*\* 넓이 : horizontalDiagonal\*verticalDiagonal / 2 ;*

*\*/*

public class Kite  extends Shape2D{

    private double horizontalDiagonal;

    private double verticalDiagonal;

    public Kite() {this(1,1);}

    public Kite(double horizontalDiagonal, double verticalDiagonal) {

        this.horizontalDiagonal = horizontalDiagonal;

        this.verticalDiagonal = verticalDiagonal;

    }

    @Override

    public void calculate() {

        System.out.println("[calculate "+this.getName()+" ]: "+this.getArea());

    }

    @Override

    public double getArea() {

*return* horizontalDiagonal\*verticalDiagonal / 2.0;

    }

*// 도형의 이름 출력*

    @Override

    public String getName() {

*return* "Kite";

    }

*// 사용자 입력 처리하기.*

    @Override

    public void getUserInput() {

        System.out.print("Please enter Kite horizontalDiagonal:");

        this.horizontalDiagonal = UserInput.getDouble();

        System.out.print("Please enter Kite verticalDiagonal:");

        this.verticalDiagonal = UserInput.getDouble();

    }

}

2.2.2.2~ 그외 Shape2D 도 동일한 구조로 구현

package Shape.Shape2Ds;

import Shape.Shape2D;

import Shape.UserInput;

*/\**

*\* 평행사변형 클래스*

*\* Paralleogram  base,height*

*\* 넓이 : base\*height*

*\*/*

public class Paralleogram  extends Shape2D{

    private double base;

    private double height;

    public Paralleogram() {this(1,1);}

    public Paralleogram(double base, double height) {

        this.base = base; this.height = height;

    }

    @Override

    public void calculate() {

        System.out.println("[calculate "+this.getName()+" ]: "+this.getArea());

    }

    @Override

    public double getArea() {

*return* base\*height;

    }

*// 도형의 이름 출력*

    @Override

    public String getName() {

*return* "Paralleogram";

    }

*// 사용자 입력 처리하기.*

    @Override

    public void getUserInput() {

        System.out.print("Please enter Paralleogram Base:");

        this.base = UserInput.getDouble();

        System.out.print("Please enter Paralleogram height:");

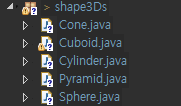
        this.height = UserInput.getDouble();

    }

}

#### 2.2.3 shape3D를 상속받아 concrete class 구현하기

콘크리트 class는 따로 package로 분리했습니다.



package Shape.shape3Ds;

import Shape.Shape3D;

import Shape.UserInput;

*/\**

*\* 원뿔 클래스*

*\* Cone  radius height*

*\* 부피 : PI \* radius\*radius\*height / 3 ;*

*\*/*

public class Cone extends Shape3D {

    private double radius;

    private double height;

    public Cone() {

        this(1,1);

    }

    public Cone(double radius, double height) {

        super();

        this.radius = radius;

        this.height = height;

    }

    @Override

    public void calculate() {

        System.out.println("[calculate "+this.getName()+" ]: "+this.getVolume());

    }

    @Override

    public double getVolume() {

*return* (Math.PI \* radius\*radius\*height)/3 ;

    }

*// 도형의 이름 출력*

    @Override

    public String getName() {

*return* "Cone";

    }

*// 사용자 입력 처리하기.*

    @Override

    public void getUserInput() {

        System.out.print("Please enter Cone radius:");

        this.radius = UserInput.getDouble();

        System.out.print("Please enter Cone height:");

        this.height = UserInput.getDouble();

    }

}

2.2.3.2~ 그외 Shape2D 도 동일한 구조로 구현

package Shape.shape3Ds;

import Shape.Shape3D;

import Shape.UserInput;

*/\**

*\* 직육면체 클래스*

*\* Cuboid  width height   length*

*\* 부피 : width\*height\*height*

*\*/*

public class Cuboid extends Shape3D {

    private double width;

    private double height;

    private double length;

    public Cuboid() {

        this(1,1,1);

    }

    public Cuboid(double width, double height,double length) {

        super();

        this.width = width;

        this.height = height;

        this.length = length;

    }

    @Override

    public void calculate() {

        System.out.println("[calculate "+this.getName()+" ]: "+this.getVolume());

    }

    @Override

    public double getVolume() {

*return* width\*height\*height;

    }

*// 도형의 이름 출력*

    @Override

    public String getName() {

*return* "Cuboid";

    }

*// 사용자 입력 처리하기.*

    @Override

    public void getUserInput() {

        System.out.print("Please enter Cuboid width:");

        this.width = UserInput.getDouble();

        System.out.print("Please enter Cuboid height:");

        this.height = UserInput.getDouble();

        System.out.print("Please enter Cuboid length:");

        this.length = UserInput.getDouble();

    }

}

#### 2.2.4 ShapeCalculator Shape2DCalculator Shape3DCalculator 구현

shapeCalculaotr

ShapeCalculator 는 팩토리 패턴에서의 핵심 추상 클래스이며,

createShape에서 타입별로 클래스틑 인스턴스화 하고

생성된 클래스를 calculate 함수에서, 사용자의 입력을 받고 실제 도형에서 calculate 한 결과를 출력합니다.

위 기능을 2D, 3D 로 나누어서 구현합니다.

package Shape;

public abstract class ShapeCalculator {

    public abstract Shape createShape(String name);

    public abstract void calculate(String name);

}

package Shape;

import Shape.Shape2Ds.Kite;

import Shape.Shape2Ds.Paralleogram;

import Shape.Shape2Ds.Rectangle;

import Shape.Shape2Ds.Square;

import Shape.Shape2Ds.Trapezoid;

import Shape.Shape2Ds.Triangle;

public class Shape2DCalculator extends ShapeCalculator {

*// Shape2DCalculator 가 가지고 있는 타입들 선언 및 리턴*

    static String[] names = {"Triangle","Kite","Paralleogram","Rectangle","Square","Trapezoid"};

    public static String[] names() {

*return* Shape2DCalculator.names;

    }

*// type에 따른 도형 생성*

    @Override

    public Shape createShape(String name) {

*if*(name == "Triangle") *return* *new* Triangle();

*if*(name == "Kite") *return* *new* Kite();

*if*(name == "Paralleogram") *return* *new* Paralleogram();

*if*(name == "Square") *return* *new* Square();

*if*(name == "Rectangle") *return* *new* Rectangle();

*if*(name == "Trapezoid") *return* *new* Trapezoid();

*return* null;

    }

*// type에 따른 도형 생성 후 userInput 후에 계산 결과 출력*

    @Override

    public void calculate(String name) {

*// shape를 생성하고*

        Shape shape = createShape(name);

*if*(shape ==null) {

            System.out.println("error : no shape");

*return*;

        }

*// shape의 getUserinput를 호출*

        shape.getUserInput();

*// shape의 calcuator를 호출한다.*

        shape.calculate();

    }

}

package Shape;

import Shape.shape3Ds.Cone;

import Shape.shape3Ds.Cuboid;

import Shape.shape3Ds.Cylinder;

import Shape.shape3Ds.Pyramid;

import Shape.shape3Ds.Sphere;

public class Shape3DCalculator extends ShapeCalculator {

*// Shape3DCalculator 가 가지고 있는 타입들 선언 및 리턴*

    static String[] names = {"Cuboid","Cylinder","Sphere","Pyramid","Cone"};

    public static String[] names() {

*return* Shape3DCalculator.names;

    }

*// type에 따른 도형 생성*

    @Override

    public Shape createShape(String name) {

*if*(name == "Cuboid") *return* *new* Cuboid();

*if*(name == "Cylinder") *return* *new* Cylinder();

*if*(name == "Sphere") *return* *new* Sphere();

*if*(name == "Pyramid") *return* *new* Pyramid();

*if*(name == "Cone") *return* *new* Cone();

*return* null;

    }

*// type에 따른 도형 생성 후 userInput 후에 계산 결과 출력*

    @Override

    public void calculate(String name) {

*// shape를 생성하고*

        Shape shape = createShape(name);

*if*(shape ==null) {

            System.out.println("error : no shape");

*return*;

        }

*// shape의 getUserinput를 호출*

        shape.getUserInput();

*// shape의 calcuator를 호출한다.*

        shape.calculate();

    }

}

#### UserInput 구현

- 실수, 정수(범위) 를 입력받습니다.

package Shape;

import java.util.Scanner;

*// scanner 클래스 static 화*

public class UserInput {

*// static 변수로 스캐너를선언*

    public static Scanner scanner = *new* Scanner(System.in);

*// double 을 입력받아서 리턴*

    public static double getDouble() {

*return* scanner.nextDouble();

    }

*// 특정 범위의 int를 입력받아서 리턴, 범위를 벗어나면 0을 출력 ( 애러 발생 고도화 가능 )*

    public static int getIntegerBetween(int a,int b) {

        int userins = scanner.nextInt();

*if*( userins >= a && userins <= b) {

*return* userins;

        }*else* {

*return* 0;

        }

    }

}

### 2.3 구현 결과

사용자로부터 모드를 입력받아.mode 에 따라서, 2d,3d 계산기를 리턴해줍니다..

해당 계산기(팩토리)가 가지고 있는 도형의 이름 가져와서, 해당 이름으로 객체를 인스턴스화 해서 계산을 해본다.

package Shape;

public class MainTest {

    public static void main(String[] args) {

        System.out.println("lab05 MAINTEST");

*// 사용자로부터 모드를 입력받아.*

        System.out.println("please enter select 1 (shape2DCalculator) or 2 (shape3DCalculator)");

        int mode = UserInput.getIntegerBetween(1, 2);

*//mode 에 따라서, 2d,3d 계산기를 리턴해준다.*

        ShapeCalculator shapeCalculator = ShapeCalculatorFactory.getShapeCalculator(mode);

*//해당 팩토리가 가지고 있는 도형의 이름 가져온다.*

        String[] names = ShapeCalculatorFactory.getShapNames(mode);

*// 해당 이름으로 객체를 인스턴스화 해서 계산을 해본다.*

*for*(String name*:*names) {

            shapeCalculator.calculate(name); *// calculate*

        }

    }

}

