

## ใบงานการทดลองที่ 8

### เรื่อง Wrapper Class และ Math Class

#### 1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการติดต่อระหว่างงาน และการเรียกใช้งานคลาสสำคัญทางคณิตศาสตร์
- 1.2. รู้และเข้าใจการใช้งานคลาส Wrapper

#### 2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

#### 3. ทฤษฎีการทดลอง

##### 3.1. Primitive Data Type คืออะไร ?

- ชนิดข้อมูลพื้นฐาน เพื่อบอกว่าเป็นข้อมูลชนิดแบบใด เช่น ตัวเลข หรือ ตัวอักษร

##### 3.2. Wrapper Class คืออะไร? มีอะไรบ้าง? อธิบายพร้อมยกตัวอย่างประกอบ

- คลาสที่ช่วยเราในการสร้างและใช้งานอ็อบเจกต์ของชนิดข้อมูลดั้งเดิมนั้นเรียกว่าคลาส java wrapper

##### 3.3. ยกตัวอย่าง Method ที่สามารถเรียกใช้งานได้ใน Wrapper Class

```
public class Wrapper {  
    private Object obj;  
    public Object getObj(){return this.obj;}  
    public void setObj(Object obj){this.obj=obj;}  
}
```

คลาส Wrapper กำหนดค่า และอ่านค่า ตามที่กำหนด แต่การอ่านค่า จะต้องแปลง (cast) ให้เป็นชนิดข้อมูลที่ต้องการและต้องตรงกับชนิดข้อมูลที่ได้กำหนดไว้

##### 3.4. คำสั่ง Math.max( x, y ) ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

- เป็นการหาค่าสูงสุดระหว่างตัวแปรหรือระหว่าง Object 2 ตัว

##### 3.5. คำสั่ง Math.min( x, y ) ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

- เป็นการหาค่าต่ำสุดระหว่างตัวแปรหรือระหว่าง Object 2 ตัว

##### 3.6. คำสั่ง Math.sqrt( x ) ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

- หารากที่สองของจำนวนที่ระบุ

##### 3.7. คำสั่ง Math.abs( x ) ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

- จะเป็นการหาค่าสัมบูรณ์ หรือค่าที่ไม่ติดลบ

##### 3.8. คำสั่ง Math.random() ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

- เป็นเมธอดที่ใช้สำหรับสุ่มตัวเลขในภาษา JavaScript

#### 4. ลำดับขั้นการปฏิบัติการ

4.1. จงพัฒนาและปรับปรุงโปรแกรมเครื่องคิดเลขจากใบงานทดลองที่ 5 โดยการเพิ่มปุ่มให้กับเครื่องคิดเลขที่มีความสามารถเพิ่มเติมดังต่อไปนี้

4.1.1. ปุ่ม sqrt เพื่อใช้ในการหาค่ารากที่ 2 ของตัวเลขที่ผู้ใช้กรอก

4.1.2. ปุ่ม abs เพื่อใช้ในการหาค่าสัมบูรณ์ของตัวเลขที่ผู้ใช้กรอก

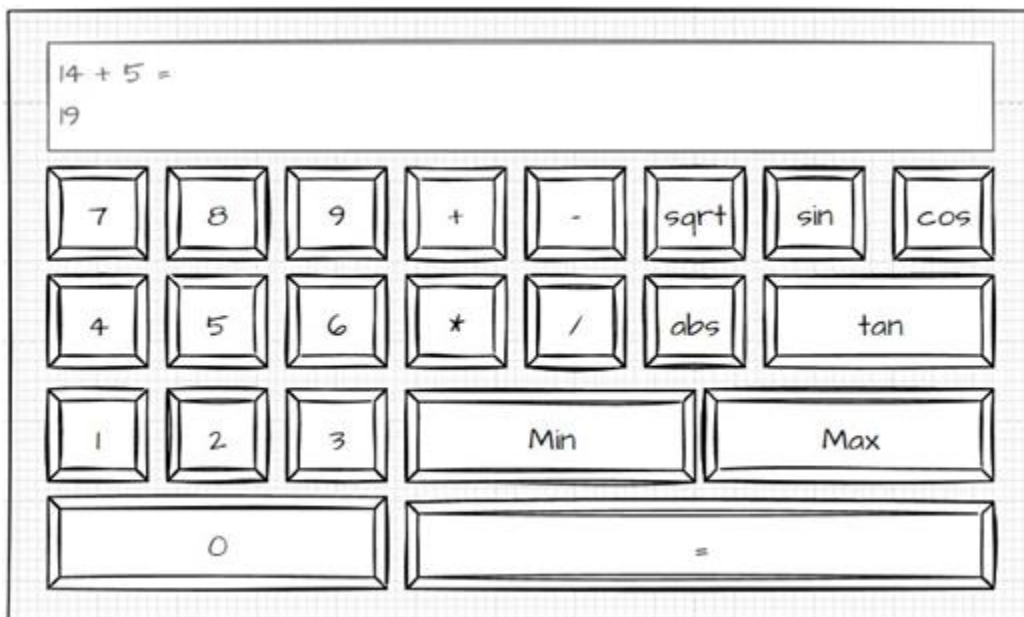
4.1.3. ปุ่ม sin เพื่อใช้ในการหา Sin ของตัวเลขที่ผู้ใช้กรอก

4.1.4. ปุ่ม cos เพื่อใช้ในการหา Cos ของตัวเลขที่ผู้ใช้กรอก

4.1.5. ปุ่ม tan เพื่อใช้ในการหา Tan ของตัวเลขที่ผู้ใช้กรอก

4.1.6. ปุ่ม Min เพื่อใช้ในการหาค่าที่น้อยที่สุดของตัวเลขที่ผู้ใช้กรอกมา 2 ค่า

4.1.7. ปุ่ม Max เพื่อใช้ในการหาค่าที่มากที่สุดของตัวเลขที่ผู้ใช้กรอกมา 2 ค่า



โค้ด โปรแกรมภายในปุ่ม เครื่องหมายเท่ากับ (เฉพาะส่วนที่เพิ่มเข้ามาในใบงานการทดลองนี้เท่านั้น)

```
Button btnTotal = new Button(panalMain, SWT.CENTER);
btnTotal.setFont(SWTResourceManager.getFont("Unispace", 78, SWT.BOLD));
btnTotal.addSelectionListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        if( answer == null ) {
            if( operation == "+" ) {
                result = Double.parseDouble(lblText1.getText()) +
Double.parseDouble(lblText2.getText()) ;
                answer = String.format("%.2f", result);
                lblShowText.setText( lblText1.getText() + " + " +
lblText2.getText() + " = " + "\n" + answer );
```

```

        } else if( operation == "-" ) {
            result = Double.parseDouble(lblText1.getText()) -
Double.parseDouble(lblText2.getText()) ;
            answer = String.format("%.2f", result);
            lblShowText.setText( lblText1.getText() + " - " +
lblText2.getText() + " = " + "\n" + answer );
        } else if( operation == "*" ) {
            result = Double.parseDouble(lblText1.getText()) *
Double.parseDouble(lblText2.getText()) ;
            answer = String.format("%.2f", result);
            lblShowText.setText( lblText1.getText() + " * " +
lblText2.getText() + " = " + "\n" + answer );
        } else if( operation == "/" ) {
            result = Double.parseDouble(lblText1.getText()) /
Double.parseDouble(lblText2.getText()) ;
            answer = String.format("%.2f", result);
            lblShowText.setText( lblText1.getText() + " / " +
lblText2.getText() + " = " + "\n" + answer );
        } else if( operation == "Min" ) {
            result =
Math.min(Double.parseDouble(lblText1.getText()), Double.parseDouble(lblText2.getText()));
            answer = String.format("%.2f", result);
            lblShowText.setText( lblText1.getText() + " Min " +
lblText2.getText() + " = " + "\n" + answer );
        } else if( operation == "Max" ) {
            result =
Math.max(Double.parseDouble(lblText1.getText()), Double.parseDouble(lblText2.getText()));
            answer = String.format("%.2f", result);
            lblShowText.setText( lblText1.getText() + " Min " +
lblText2.getText() + " = " + "\n" + answer );
        } else if( operation == "sqrt" ) {
            result = Math.sqrt(Double.parseDouble(
lblText1.getText() ));
            answer = String.format("%.2f", result);
            lblShowText.setText( answer );
            tempStr = "0" ;
        } else if( operation == "abs" ) {
            result = Math.abs(Double.parseDouble(
lblText1.getText() ));
            answer = String.format("%.2f", result);
            lblShowText.setText( tempStr2 + " = " + "\n" + answer
);
            tempStr = "0" ;
        } else if( operation == "sin" ) {
            result = Math.sin(
Double.parseDouble(lblText1.getText()) );
            answer = String.format("%.4f", result);

```

```

        lblShowText.setText( tempStr2 + " = " + "\n" +
answer );
        tempStr = "0" ;
    } else if( operation == "cos" ) {
        result = Math.cos(
Double.parseDouble(lblText1.getText()) );
        answer = String.format("%.4f", result);
        lblShowText.setText( tempStr2 + " = " + "\n" +
answer );
        tempStr = "0" ;
    } else if( operation == "tan" ) {
        result = Math.tan(
Double.parseDouble(lblText1.getText()) );
        answer = String.format("%.4f", result);
        lblShowText.setText( tempStr2 + " = " + "\n" +
answer );
        tempStr = "0" ;
    } //end else if
    } else {
        if( operation == "+" ) {
            tempAnswer = answer;
            result = Double.parseDouble(answer) +
Double.parseDouble(lblText2.getText()) ;
            answer = String.format("%.2f", result);
            lblShowText.setText( tempAnswer + " + " +
lblText2.getText() + " = " + "\n" + answer );
        } else if( operation == "-" ) {
            tempAnswer = answer;
            result = Double.parseDouble(answer) -
Double.parseDouble(lblText2.getText()) ;
            answer = String.format("%.2f", result);
            lblShowText.setText( tempAnswer + " - " +
lblText2.getText() + " = " + "\n" + answer );
        } else if( operation == "*" ) {
            tempAnswer = answer;
            result = Double.parseDouble(answer) *
Double.parseDouble(lblText2.getText()) ;
            answer = String.format("%.2f", result);
            lblShowText.setText( tempAnswer + " * " +
lblText2.getText() + " = " + "\n" + answer );
        } else if( operation == "/" ) {
            tempAnswer = answer;
            result = Double.parseDouble(answer) /
Double.parseDouble(lblText2.getText()) ;
            answer = String.format("%.2f", result);
            lblShowText.setText( tempAnswer + " / " +
lblText2.getText() + " = " + "\n" + answer );
        } else if( operation == "Min" ) {

```

```

        tempAnswer = answer;
        result = Math.min( Double.parseDouble(answer) ,
        Double.parseDouble(lblText2.getText()) );
        answer = String.format("%.2f", result);
        lblShowText.setText( tempAnswer + " Min " +
        lblText2.getText() + " = " + "\n" + answer );
    }else if( oparation == "Max" ) {
        tempAnswer = answer;
        result = Math.max( Double.parseDouble(answer) ,
        Double.parseDouble(lblText2.getText()) );
        answer = String.format("%.2f", result);
        lblShowText.setText( tempAnswer + " Max " +
        lblText2.getText() + " = " + "\n" + answer );
    }else if( oparation == "sqrt" ) {
        tempAnswer = answer;
        result = Math.sqrt( Double.parseDouble( answer ) );
        answer = String.format("%.2f", result);
        lblShowText.setText( answer );
        tempStr = "0" ;
    }else if( oparation == "abs" ) {
        tempAnswer = answer;
        result = Math.abs( Double.parseDouble( answer ) );
        answer = String.format("%.2f", result);
        lblShowText.setText( tempStr2 + " = " + "\n" + answer
        );
        tempStr = "0" ;
    }else if( oparation == "sin" ) {
        tempAnswer = answer;
        result = Math.sin( Double.parseDouble( answer ) );
        answer = String.format("%.4f", result);
        lblShowText.setText( tempStr2 + " = " + "\n" +
        answer );
        tempStr = "0" ;
    }else if( oparation == "cos" ) {
        tempAnswer = answer;
        result = Math.cos( Double.parseDouble( answer ) );
        answer = String.format("%.4f", result);
        lblShowText.setText( tempStr2 + " = " + "\n" +
        answer );
        tempStr = "0" ;
    }else if( oparation == "tan" ) {
        tempAnswer = answer;
        result = Math.tan( Double.parseDouble( answer ) );
        answer = String.format("%.4f", result);
        lblShowText.setText( tempStr2 + " = " + "\n" +
        answer );
        tempStr = "0" ;
    }//end else if

```

```

        }// end else
        //lblText1.setText("");
        lblText2.setText("");
        StateOp = false;
    }// end method
});
btnTotal.setText("=");
btnTotal.setBounds(345, 212, 193, 114);
btnTotal.setBackground( new Color(240, 128, 128) );
Button btnMin = new Button(panalMain, SWT.NONE);
btnMin.setFont(SWTResourceManager.getFont("Unispace", 11, SWT.BOLD));
btnMin.addSelectionListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        if( tempStr != "" ) {
            oparation="Min";
            StateOp = true;
            if( answer == null ) {
                tempStr2 = tempStr+oparation;
                Stroparation = tempStr2 + btnMin.getText();
                tempStr = "";
                lblShowText.setText(tempStr2);
            } else {
                tempStr2 = answer + oparation;
                Stroparation = answer + btnMin.getText();
                tempStr = "";
                lblShowText.setText(tempStr2);
            }
        }// end else
    } //end if
} //end method
});
btnMin.setBounds(211, 212, 128, 54);
btnMin.setText("Min");

Button btnMax = new Button(panalMain, SWT.NONE);
btnMax.setFont(SWTResourceManager.getFont("Unispace", 11, SWT.BOLD));
btnMax.addSelectionListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        if( tempStr != "" ) {
            oparation="Max";
            StateOp = true;
            if( answer == null ) {
                tempStr2 = tempStr+oparation;
                Stroparation = tempStr2 + btnMin.getText();
                tempStr = "";
                lblShowText.setText(tempStr2);
            } else {

```

```

        tempStr2 = answer + oparation;
        Stroparation = answer + btnMin.getText();
        tempStr = "";
        lblShowText.setText(tempStr2);
    } // end else
} //end if
} // end method
});
btnMax.setText("Max");
btnMax.setBounds(211, 272, 128, 54);

Button btnabs = new Button(panalMain, SWT.NONE);
btnabs.setFont(SWTResourceManager.getFont("Unispace", 11, SWT.BOLD));
btnabs.addSelectionListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        if( tempStr != "" ) {
            oparation="abs";
            StateOp = true;
            if( answer == null ) {
                tempStr2 = tempStr + oparation;
                Stroparation = tempStr2 + btnabs.getText();
                tempStr = "";
                lblShowText.setText( tempStr2 );
            } else {
                tempStr2 = answer + oparation;
                Stroparation = answer + btnabs.getText();
                tempStr = "";
                lblShowText.setText( tempStr2 );
            }
        } //end else
    } //end if
} // end method
});
btnabs.setText("abs");
btnabs.setBounds(345, 152, 61, 54);

Button btnSqrt = new Button(panalMain, SWT.NONE);
btnSqrt.setFont(SWTResourceManager.getFont("Unispace", 11, SWT.BOLD));
btnSqrt.addSelectionListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        if( tempStr != "" ) {
            oparation="sqrt";
            StateOp = true;
            if( answer == null ) {
                tempStr2 = tempStr + oparation;
                Stroparation = tempStr2 + btnSqrt.getText();
                tempStr = "";
            }
        }
    }
}

```

```

        lblShowText.setText( tempStr2 );
    } else {
        tempStr2 = answer + operation;
        Stroparation = answer + btnSqrt.getText();
        tempStr = "";
        lblShowText.setText( tempStr2 );
    } //end else
} //end if
} // end method
});
btnSqrt.setText("Sqrt");
btnSqrt.setBounds(345, 92, 61, 54);

Button btnSin = new Button(panalMain, SWT.NONE);
btnSin.setFont(SWTResourceManager.getFont("Unispace", 11, SWT.BOLD));
btnSin.addSelectionListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        if( tempStr != "" ) {
            operation="sin";
            StateOp = true;
            if( answer == null ) {
                tempStr2 = operation + "(" + tempStr + " )";
                Stroparation = tempStr2 + btnSin.getText();
                tempStr = "";
                lblShowText.setText( tempStr2 );
            } else {
                tempStr2 = operation + "(" + answer + " )";
                Stroparation = answer + btnSin.getText();
                tempStr = "";
                lblShowText.setText( tempStr2 );
            } //end else
        } //end if
    } // end method
});
btnSin.setText("sin");
btnSin.setBounds(412, 92, 61, 54);

Button btnTan = new Button(panalMain, SWT.NONE);
btnTan.setFont(SWTResourceManager.getFont("Unispace", 11, SWT.BOLD));
btnTan.addSelectionListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        if( tempStr != "" ) {
            operation="tan";
            StateOp = true;
            if( answer == null ) {
                tempStr2 = operation + "(" + tempStr + " )";

```



```

        Stroparation = tempStr2 + btnTan.getText();
        tempStr = "";
        lblShowText.setText( tempStr2 );
    } else {
        tempStr2 = oparation + "(" + answer + ")" ;
        Stroparation = answer + btnTan.getText();
        tempStr = "";
        lblShowText.setText( tempStr2 );
    } //end else
} //end if
} // end method

});
btnTan.setText("tan");
btnTan.setBounds(412, 152, 128, 54);

Button btnCos = new Button(panalMain, SWT.NONE);
btnCos.setFont(SWTResourceManager.getFont("Unispace", 11, SWT.BOLD));
btnCos.addSelectionListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        if( tempStr != "" ) {
            oparation="cos";
            StateOp = true;
            if( answer == null ) {
                tempStr2 = oparation + "(" + tempStr + ")" ;
                Stroparation = tempStr2 + btnCos.getText();
                tempStr = "";
                lblShowText.setText( tempStr2 );
            } else {
                tempStr2 = oparation + "(" + answer + ")" ;
                Stroparation = answer + btnCos.getText();
                tempStr = "";
                lblShowText.setText( tempStr2 );
            } //end else
        } //end if
    } // end method

});
btnCos.setText("cos");
btnCos.setBounds(479, 92, 61, 54);
}

```

## 5. สรุปผลการปฏิบัติการ

## 6. คำถามท้ายการทดลอง

### 6.1. Wrapper Class ควรใช้เมื่อใด? เพราะอะไร?

- เราสามารถใช้ดำเนินการกับชนิดข้อมูลแบบ primitive data type ได้เช่นเดียวกับการใช้งานคลาสคือมีเมธอดให้เรียกใช้งานเพื่อให้ผลลัพธ์ในรูปแบบต่างๆ เช่น แปลงชนิดข้อมูล แปลงรูปแบบของข้อมูล แก้ไขข้อมูลในรูปแบบต่างๆ

### 6.2. ข้อควรระวังในการใช้ Math Class คืออะไร?

- มีเมธอดหลายๆเมธอดให้เลือกใช้เพื่อจัดการกับข้อมูล เช่น หากเรามีข้อมูลตัวเลขในรูปแบบของชนิดข้อมูลแบบ String และเราต้องการเอามาคำนวน เราจะต้องแปลงชนิดข้อมูลดังกล่าวก่อน ซึ่งเราไม่สามารถทำได้กับชนิดข้อมูลแบบ primitive data type แต่เราสามารถใช้ Wrapper Class ดำเนินการได้ การเรียกใช้ Wrapper Class เหมือนการกำหนดให้ข้อมูลแบบ primitive data type เป็นพารามิเตอร์ของ Wrapper Class และเราจะดำเนินการกับฟิลด์ด้วยเมธอดของ Wrapper Class