

2024학년도 1학기
JAVA 프로그래밍 I

514760
2024년 봄학기
5/9/2024
박경신

과제 Lab4 (Inheritance & Interface)

- Lab3 프로그램을 Inheritance와 Interface를 활용한다.
- **BlockType 열거형**
 - `import java.awt.Color;`
 - `RED_BLOCK(1, Color.RED)`
 - `TEAL_BLOCK(2, new Color(54, 117, 136))`
 - `DOGERS_BLUE_BLOCK(3, new Color(0, 90, 156))`
 - `private int value;`
 - `private Color color;`
 - Constructor 구현
 - Getter 구현

과제 Lab4 (Inheritance & Interface)

□ IMoveable 인터페이스

- *void translate(double x, double y); 추상메소드 선언*

□ IColliding 인터페이스

- *boolean collideWith(Block other); 추상메소드 선언*

과제 Lab4 (Inheritance & Interface)

□ RandomUtils 클래스

- `public static BlockType randomBlockType()` 내부 구현
- `public static int randomInt(int min, int max)` 내부 구현
- `public static double randomDouble(double min, double max)` 내부 구현

□ BlockFactory 클래스

- `public static Block getInstance(BlockType type, double x, double y)` 내부 구현

과제 Lab4 (Inheritance & Interface)

□ Block 추상클래스는 IMoveable과 IColliding 을 상속

- `import java.awt.geom.Rectangle2D;`
- `import java.awt.Graphics2D;`
- `import java.awt.Color;`

- `protected BlockType type;`
- `protected Rectangle2D[] rectangles = new Rectangle2D[4];`
- `protected Block(double x, double y) { init(x, y); }`
- Getter
- *`public abstract void init(double x, double y); 추상메소드 선언`*
- `public void translate(int x, int y)` block의 4개 `rect.setRect` 내부 구현 필요함
- `public void collideWith(Block other)` block의 4개 `rect.intersects` 내부 구현 필요함
- `public String toString()` 내부 구현 필요함

과제 Lab4 (Inheritance & Interface)

□ Block 추상클래스는 IMoveable과 IColliding 을 상속

■ public void draw(Graphics2D g2) 내부 구현

```
public void draw(Graphics2D g2) { // draw based on init rectangles
    for (var rect : rectangles) {
        // fill with blockType color
        g2.setPaint(type.getColor());
        g2.fill(rect);
        // draw BLACK border line
        g2.setPaint(Color.BLACK);
        g2.draw(rect);
    }
}
```

과제 Lab4 (Inheritance & Interface)

□ RedBlock/TealBlock/DogersBlueBlock 클래스는 Block 추상클래스 상속

■ 생성자 내부 구현

■ public void init(double x, double y) 내부 구현

@Override

```
public void init(double x, double y) { // RedBlock init 내부구현
    rectangles[0] = new Rectangle2D.Double(x, y, BLOCK_WIDTH, BLOCK_WIDTH);
    rectangles[1] = new Rectangle2D.Double(x + BLOCK_WIDTH, y, BLOCK_WIDTH,
BLOCK_WIDTH);
    rectangles[2] = new Rectangle2D.Double(x, y + BLOCK_WIDTH, BLOCK_WIDTH,
BLOCK_WIDTH);
    rectangles[3] = new Rectangle2D.Double(x + BLOCK_WIDTH, y + BLOCK_WIDTH,
BLOCK_WIDTH, BLOCK_WIDTH);
}
```

과제 Lab4 (Interface & Collection)

□ BlockManager 클래스 제공

```
class BlockManager extends JPanel implements ActionListener {
    private Timer timer = null;
    private final static int DELAY = 500; // 500 msec delay
    private Block[] blocks = new Block[4];
    public BlockManager() { // Constructor
        timer = new Timer(DELAY, this); // timer ActionListener event (500ms DELAY)
        this.setDoubleBuffered(true);
        blocks[0] = BlockFactory.getInstance(BlockType.RED_BLOCK, 50.0, 0.0);
        blocks[1] = BlockFactory.getInstance(BlockType.TEAL_BLOCK, 180.0, 10);
        blocks[2] = BlockFactory.getInstance(BlockType.DOGERS_BLUE_BLOCK, 70.0, 170.0);
        blocks[3] = BlockFactory.getInstance(RandomUtils.randomBlockType(),
        RandomUtils.randomDouble(200.0,400.0), RandomUtils.randomDouble(20.0,40.0));
    }
    public void translate(int index, double dx, double dy) { // translate
        blocks[index].translate(dx, dy);
    }
}
```


과제 Lab4 (Interface & Collection)

```
public void collide() { // collide
    for (Block block : blocks) {
        for (Block otherBlock : blocks) {
            if (block != otherBlock && block.collideWith(otherBlock)) {
                System.out.println(block.getType() + " collide with " + otherBlock.getType());
            }
        }
    }
}

@Override
public void paintComponent(Graphics g) {
    super.paintComponent(g); // JPanel paintComponent
    Graphics2D g2 = (Graphics2D) g;
    for (var block : blocks) { // RED_BLOCK, TEAL_BLOCK, DOGERS_BLUE_BLOCK, RANDOM_BLOCK
        if (block != null) block.draw(g2);
    }
}
```

과제 Lab4 (Interface & Collection)

@Override

```
public void actionPerformed(ActionEvent e) {
    for (int i = 0; i < blocks.length; i++) {
        translate(i, RandomUtils.randomInt(-1, 2), RandomUtils.randomInt(1, 5));
    }
    collide(); // check collide
    repaint(); // repaint
}

public void start() { // timer start
    if (timer.isRunning())
        timer.restart();
    else
        timer.start();
}

public void stop() {
    timer.stop();
}
```

과제 Lab4 (Interface & Collection)

□ Lab4MainFrame 클래스 제공

```
public class Lab4MainFrame extends JFrame implements KeyListener {  
    private JPanel displayPanel = null;  
    private BlockManager blockManager = null;  
    private int selectedIndex = 0;  
    public Lab4MainFrame(String title) {  
        super(title);  
        blockManager = new BlockManager();  
        blockManager.start(); // blockManager timer start  
        displayPanel = new JPanel(new BorderLayout());  
        displayPanel.add(blockManager, BorderLayout.CENTER);  
        displayPanel.addKeyListener(this); // KeyListener  
        displayPanel.setFocusable(true);  
        displayPanel.requestFocusInWindow();  
        this.getContentPane().add(displayPanel);  
        this.setSize(600, 600);  
        this.setVisible(true);  
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    }  
}
```

과제 Lab4 (Interface & Collection)

@Override

```
public void keyPressed(KeyEvent e) {
    if (e.getKeyCode() == KeyEvent.VK_RIGHT) { // handle right arrow key
        blockManager.translate(selectedIndex, 5.0, 0.0); // move to the right
    } else if (e.getKeyCode() == KeyEvent.VK_LEFT) { // handle left arrow key
        blockManager.translate(selectedIndex, -5.0, 0.0); // move to the left
    } else if (e.getKeyCode() == KeyEvent.VK_UP) { // handle up arrow key
        blockManager.translate(selectedIndex, 0.0, -5.0); // move up (y decreases!)
    } else if (e.getKeyCode() == KeyEvent.VK_DOWN) { // handle down arrow key
        blockManager.translate(selectedIndex, 0.0, 5.0); // move down (y increases!)
    } ... // 중간생략
    blockManager.collide(); // blockManager check collide
    displayPanel.revalidate(); // repaint
    displayPanel.repaint(); // repaint
}

public static void main(String[] args) {
    new Lab4MainFrame("Block");
}
}
```

과제 제출

- Lab4와 보고서 전체를 묶어서 e-learning에 과제 제출 (due by 5/22)
 - 본인이 원하는 코드 추가 구현 및 테스트
 - Block이 collide 했을 시 더 이상 translate 하지 않음
 - Block이 bottom line을 지났을 시 더 이상 translate 하지 않음
 - 보고서에 전체 코드 분석 및 구현 내용 자세히 설명