Java20-2-Lab4

32191826 박주희

<프로그램의 전체적인 구조>

Tshirt의 종류🡪 short, long, sweat, hooded

TshirtsDecorator의 종류🡪 Heart, Tree, Text

Jcombobox로 Tshirt의 종류를 선택한 후 JButton으로 Tshirt를 decorate하는 프로그램

<구현 방법>

코드를 짜기전, 수업시간에 배웠던 Decorater 예제를 먼저 생각해보았다. 그리고 Decorater 예제에 있던 클래스와 메소드들을 Lab4와 매치해보았다.

Baverage=Tshirt

커피의 종류=티셔츠의 종류

Condiment=TshirtDecorator

커피에 올라가는 것들(휘핑, 우유 등)=티셔츠를 꾸미는 것들

수업시간에 배웠던 예제로 생각하니 프로그램을 짜기에 한결 쉬었다.

<구현 과정>

어려움이 있었던 메소드는 Decorator 클래스들의 paintComponent(Graphics g)였다.

교수님께서 설명해주신대로

g2.drawImage(**super**.decoratedTshirt.image, 0, 0, ***WIDTH***, ***HEIGHT***, **this**);

g2.drawImage(logo,225,225,***WIDTH***/4, ***HEIGHT***/4, **this**);

tshirt를 draw하고 logo를 draw하였는데 처음으로 heart버튼을 누르면 잘 작동하는데 다른 버튼을 또 누르면 그 위에 logo가 그려지지 않고 티셔츠와 heart가 모두 사라져버린채로 새로운 버튼의 logo만 떴다.

생각해보니 MainFrame에서는 panel을 계속 reset하기 때문에 기존의 logo들을 저장한 티셔츠 자체를 새로 draw해야된다는 것을 깨달았다.

그러다가 getDescription()과 cost()메소드에서는 기존의 정보들을 저장한 값을 return한다는 것을 알고 **super**.decoratedTshirt 를 이용해야겠다고 생각하였다.

**super**.decoratedTshirt.paintComponent(g);

그리고 이 한줄을 넣으니 제대로 작동되는 것을 알 수 있었다.

<Main class의 주요 메소드>



button1,2,3는 각각 heart, tree, text decorator를 나타낸다.

displayPanel.remove(product)를 함으로서 기존의 그림 자체를 지워버리고

product에 각각 해당하는 decorator 객체를 생성한 것을 저장한 후 새롭게 그림을 그린다.



comboboxItem배열에는 Tshirt종류의 객체들이 존재한다.

combobox에서 무엇을 선택했느냐에 때라 product가 바껴야하므로 조건문을 이용한다.

<My code>



TextDecorator의 settext() 메소드: 선택한 color를 return

Graphics의 setColor와 java.awt.Color를 이용하였다.

count 변수를 이용하여 text 후 heart를 눌렀을 때 또 다시 color을 선택하지 않도록 하였다.

처음엔 조건문 안에 setColor를 넣었으나 그렇게 되면 text 후 heart를 눌렀을 때 setColor가 적용되지 않음을 알고 밖으로 뺴주었다.

<결과창>

