

1) 프로젝트 개요

a) 내가 겪고 있는 문제 상황

부모님이 자영업을 하시는데 포스기(주문을 입력하고 결제하는 기기)를 임대하려면 매달 많은 금액을 지불하셔야 한다고 합니다. 그래서 저희 가게에서는 포스기를 사용하지 않고 컴퓨터 메모장을 이용해 주문을 받고 있습니다. 하지만 메모장은 GUI가 아니기 때문에 주문 내역이 한 눈에 들어오지 않고 주문을 추가하는 것도 매우 번거롭습니다.

b) 문제 상황을 해결할 수 있는 방법(2 가지 이상)

1. 많은 돈을 지불하고 포스기를 임대하는 방법
2. 수기로 주문 목록을 작성하는 방법
3. GUI 프로그램을 활용하는 방법

c) 여러가지 해결 방법 중 내가 선택한 방법

3. GUI 프로그램을 활용하는 방법

이유 : 포스기를 임대받는 비용은 소규모 자영업자의 입장에서 많은 부담이 됩니다. 하지만 수기로 일일이 주문을 받기에는 인쇄에 필요한 종이와 잉크의 값도 무시하지 못하며 쓰레기 처리도 상당히 번거롭습니다. 또한 바쁜 경우에는 모든 주문을 수기로 적는 것은 시간이 오래 걸리고 불편할 것입니다.

d) 예상되는 모습, 기대하는 효과

포스기를 임대하지 않고도 포스기와 같은 역할을 할 수 있는 프로그램은 비용적인 측면에서 지출을 줄일 수 있으며 GUI를 사용했기 때문에 바쁜 경우에도 마우스 클릭 2~3 번 안에 주문할 수 있어 시간 부담도 줄어들 것입니다.

2) 프로젝트 계획

a) 주차별 작업 계획(계획)

초기의 프로젝트를 위해 공부를 하던 중 도저히 공부한 양으로는 원하는 프로그램을 제작할 수 없을 거란 생각이 들어 23일부터 좀 더 쉬운

프로젝트를 진행하기로 결정했습니다. 따라서 11 월 23 일-30 일에는 프로그램의 그래픽 요소와 기본적인 틀을 만들기로 계획했고 12 월 1 일-12 일에 이벤트 처리를 하기로 계획을 세웠습니다.

b) 주차별 작업 계획(실제 진행)

하지만 계획과는 달리 컴퓨터 프로그래밍 2 팀 프로젝트를 준비하고 군 지원에 대해 알아보고 지원하는 데 시간을 많이 소모하여 첫 주차에는 거의 작업을 하지 못했습니다. 그래서 2 주차에 하루에 2 시간 이상씩 프로그램을 짜고 제대로 작동하지 않는 부분을 고치는 것을 반복했습니다. 2 주차 1-3 일에는 컴포넌트를 생성하고 클릭 시 다른 화면으로 넘어가는 기능을 구현했으며 3-10 일에는 메뉴를 주문하고 주문 내역을 저장하고 출력하는 등 주문과 관련한 이벤트를 처리하는 데 집중했습니다. 12 일에는 파일 입출력을 활용하여 결제 정보를 텍스트 파일로 저장하는 기능을 추가했습니다.

3) 프로젝트 결과

a) 계획서 대비 어떤 부분을 추가, 수정 하였는지 기술

계획서와는 완전히 다른 프로젝트를 진행하였기 때문에 모든 걸 기술할 수는 없을 것 같습니다.

b) 깃허브 링크 주소

<https://github.com/1547892user/Computer-Programming2-Personal-Project>

c) 동작 데모 영상 링크 주소

<https://youtu.be/RWqvjKPnUcc>

4) 회고

a) 회고는 다음 페이지에 작성하였습니다.

5) 참고한 자료

<https://www.javacodex.com/More-Examples/2/6>

마우스 클릭 버튼을 알아오는 예제

<https://blog.naver.com/clonevader/70007418173>

SetpreferredSize 예제

<https://stackoverflow.com/questions/1097366/java-swing-revalidate-vs-repaint>

revalidate() 메소드

회고

우선 프로젝트를 진행하면서 느낀 점은 자바 코딩은 골치가 아프다는 것이었습니다. 여태까지는 C 언어로 대부분 단순한 프로그램만을 만들었고 약간 복잡한 경우에도 C 언어의 자유도 덕분에 원하는 기능을 구현하기가 쉬웠습니다. 하지만 자바의 경우에는 객체를 생성하고, GUI 에 컴포넌트를 달고, 컴포넌트에 이벤트 리스너를 달고, 정보를 저장하고, 객체에서 정보를 불러오는 과정이 모두 객체 생성 이후, 그리고 필드와 메소드를 이용하여 접근해야 하기 때문에 하나의 정보에 접근하기 위해서도 그 과정이 상당히 복잡했습니다. 예를 들어 콘텐츠팬에 달려 있는 JButton 의 이벤트 리스너에서 JFrame 에 존재하는 어떤 변수를 참조하려면 JFrame 에서는 필드 멤버로 선언되어 있어야 했으며 JFrame 객체를 참조하는 변수를 만들고 getTopLevelAncestor()메소드로 참조하도록 해야 했습니다. 그냥 메모리에 직접 쓰고 포인터나 변수를 사용하여 데이터를 참조하는 C 언어와 너무나도 다르고 복잡했기 때문에 프로젝트를 진행하며 머리가 아픈 적이 꽤 있었습니다. 또한 프로젝트 진행은 필요한 기능을 하나하나 추가하는 방식으로 진행되었는데 이것 또한 좋은 방법이 아닌 것 같다는 생각이 들었습니다. 보통 일을 진행하려면 전체적인 계획을 세우고 시작하기 마련입니다. 하지만 이번 프로젝트 진행의 경우 전체적인 계획을 세우기는 했지만 구체적으로 어떻게 실행할지는 모른 채 일단 해보자 하는 식으로 시작했습니다. 그런데 진행하다 보니 문제가 하나 생기고 한 문제를 해결했더니 또 다른 문제가 생겨 문제를 연이어 처리하는 과정에서 제가 짠 코드를 계속 쳐다보고 있어야하는 부작용이 있었습니다. 또한 같은 기능을 하는 라인을 계속 반복하여 쓰는 것도 더 간단하게 작성할 수 있는 방법이 있을 것 같다는 느낌이 들었습니다. 아마도 이런 문제는 프로그램 구조를 제대로 세우지 못하고 객체지향 프로그래밍 활용을 제대로 하지 못하는 저의 능력 부족으로 발생했다고 생각합니다. 따라서 향후에 프로그래밍을 다시 하게 된다면 무작정 코드를 작성하고 점검하며 진행하기보다는 전체적인 프로그램 구조가 명확해질 때까지 계속 반복하여 생각한 뒤 프로그램을 작성하며 발생하는 부수적인 문제들만 처리할 수 있도록 코딩을 해 볼 계획입니다.