

심플한 계산기 프로젝트

[통계](#) [수정](#) [삭제](#)

namwlee99 · 약 1시간 전

 0

1. 1일차

1. 1차 회의(30분)

1. 팀장 결정

가위바위보로 결정

팀장: 이남웅 팀원: 김정관, 배용진

2. 요구 분석

1. 새로운 git repo 생성 필요 -> github를 통해 simple_calculation_project을 생성
2. UI 구성 필요 -> 위에는 결과 출력 화면을 크게 구성, 아래에는 숫자 입력 화면과 연산 기호 버튼으로 구성
3. 텍스트 입출력 구현, 버튼 구현 -> 결과 출력 화면은 입력을 못하게 하고, 입력하면에서 입력을 하였을 때 결과 출력화면과 함께 출력 되도록, 연산 기호 버튼을 눌렀을 때에는 연산 기호가 숫자 입력 화면에 입력이 되도록
4. 스플릿 구현, 연산 구현 -> 식을 입력 받으면 숫자와 연산 기호를 구분할 스플릿 함수 구현 및 숫자와 연산 기호를 구분한 배열을 계산한 연산 함수 구현

3. 업무 분당

1. 김정관
연산 버튼 구현 및 입력, -기호 붙이기/빼기 함수 구현

2. 배용진

연산식 스플릿 및 연산 함수 구현

3. 이남웅

UI 구성 및 기타 사항 구성

4. 구현 시간 결정

1시간

2. 2차 회의(30분)

40% 구현

1. 개발 현황 및 어려운 점

김정관

- 1차 회의때 프로젝트 로직에 대한 이해 부족으로 인해 다른 연산 기능 구현
1차 회의 때에는 공학용 계산기 방식으로 숫자와 연산 기호가 쌓이는 방식으로 방향을 잡았지만 김정관님은 간단한 계산기 방식으로 2개의 연산자만 계산할 수 있는 계산기 방식을 구현
-> 첫번째 목표가 공학용 계산기 방식으로 방향을 잡았던 만큼 공학용 계산기 방식으로 방향을 결정
- DEL 버튼의 지우기 기능과 C버튼의 초기화 기능 구현 완료

배용진

- 연산식 스플릿 구현 완료
- 스플릿 구현을 마치고 연산기능을 구현 중 공학용 계산기 방식이 3개 이상의 연산을 지원하는 만큼 다중 연산 기능을 어떻게 구현할 지 어려움을 느낌
-> 일단을 초기 목표로 2개의 연산자만 계산할 수 있는 방식으로 잡고, 이후 3개이상의 연산을 지원하는 계산기 방식으로 바꾸는 것을 결정

이남웅

- UI구현 완료
- git branch를 생성하고 main branch로 이동하여 커밋을 하려고 했지만 main branch를 이동하면서 visual studio가 main branch로 이동하면서 코드도 동시에 바뀌어 버려 코드가 사라짐
-> visual studio의 변경사항 보고를 통해 코드 복구

2. 업무 분담

김정관

1. 기존 목표였던 연산 버튼 구현 및 -기호 붙이기/빼기 버튼 구현

배용진

1. 연산 기능 구현

이남웅

1. git 문제 해결
2. 입력 예외처리 구현

2. 2일차

1. 3차 회의(30분)

50% 구현

1. 개발 현황 및 어려운 점

김정관

1. 로그 기능을 집에서 구현해보았지만 이전의 간단한 계산기 방식으로 구현되어있음
-> 계산식을 통째로 저장하여 log버튼을 눌렀을 때 이전 계산식이 최대 10개까지 나오도록 구현

배용진

1. 연산 기호 버튼 구현 완료
2. 연산 기능이 되는 함수를 찾아서 3개이상의 연산자까지도 구현이 가능할 듯 함
-> 3개이상의 연산자가 가능한 함수 구현

이남웅

1. 특이사항 없음

업무 분담

김정관

1. 계산식 로그 기능 구현

배용진

1. 다중 연산 기능 구현

이남웅

1. 예외 입력 처리 구현

3. 구현 시간 결정

1시간 30분

2. 4차 회의

90% 구현

1. 개발 현황 및 어려운 점

김정관

1. 계산식 로그 기능 구현 완료

배용진

1. 다중 연산 기능 구현 완료
2. 연산 입력 버튼을 할 때 연산 기호가 존재할 경우 연산 기호가 입력이 안되도록 수정중
하지만 숫자를 입력하고 연산 기호를 넣을 때 2개의 연산기호가 들어가는 등 예외상황이
많음
-> 만약 계산식이 잘못된 경우 Error표시가 나도록 결정

이남웅

1. 예외 입력 처리 구현

2. 업무 분당

김정관, 배용진

1. 각자 구현한 코드를 합치고 테스트까지 완료

이남웅

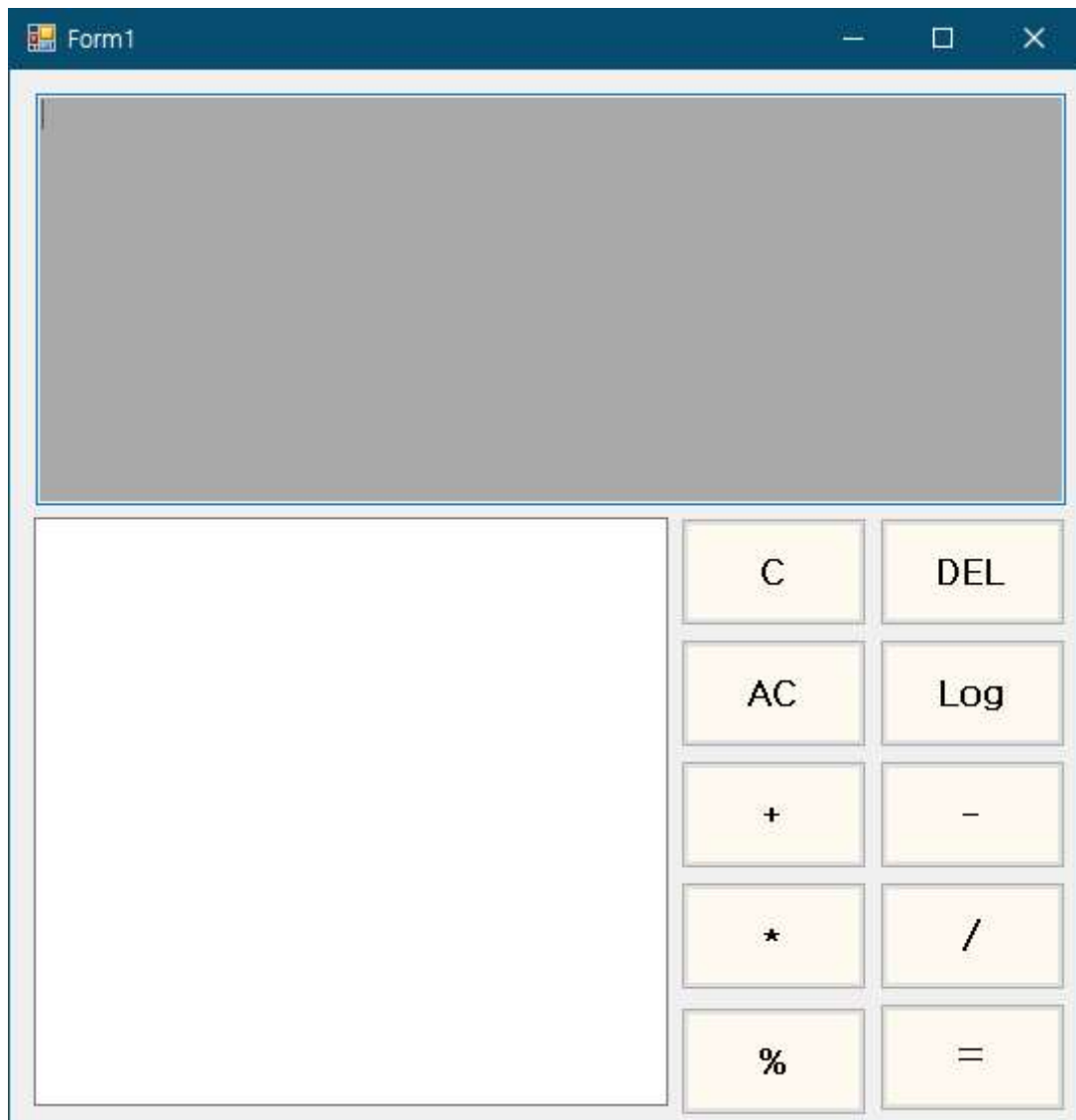
1. 서류 처리

3. 구현 시간 결정

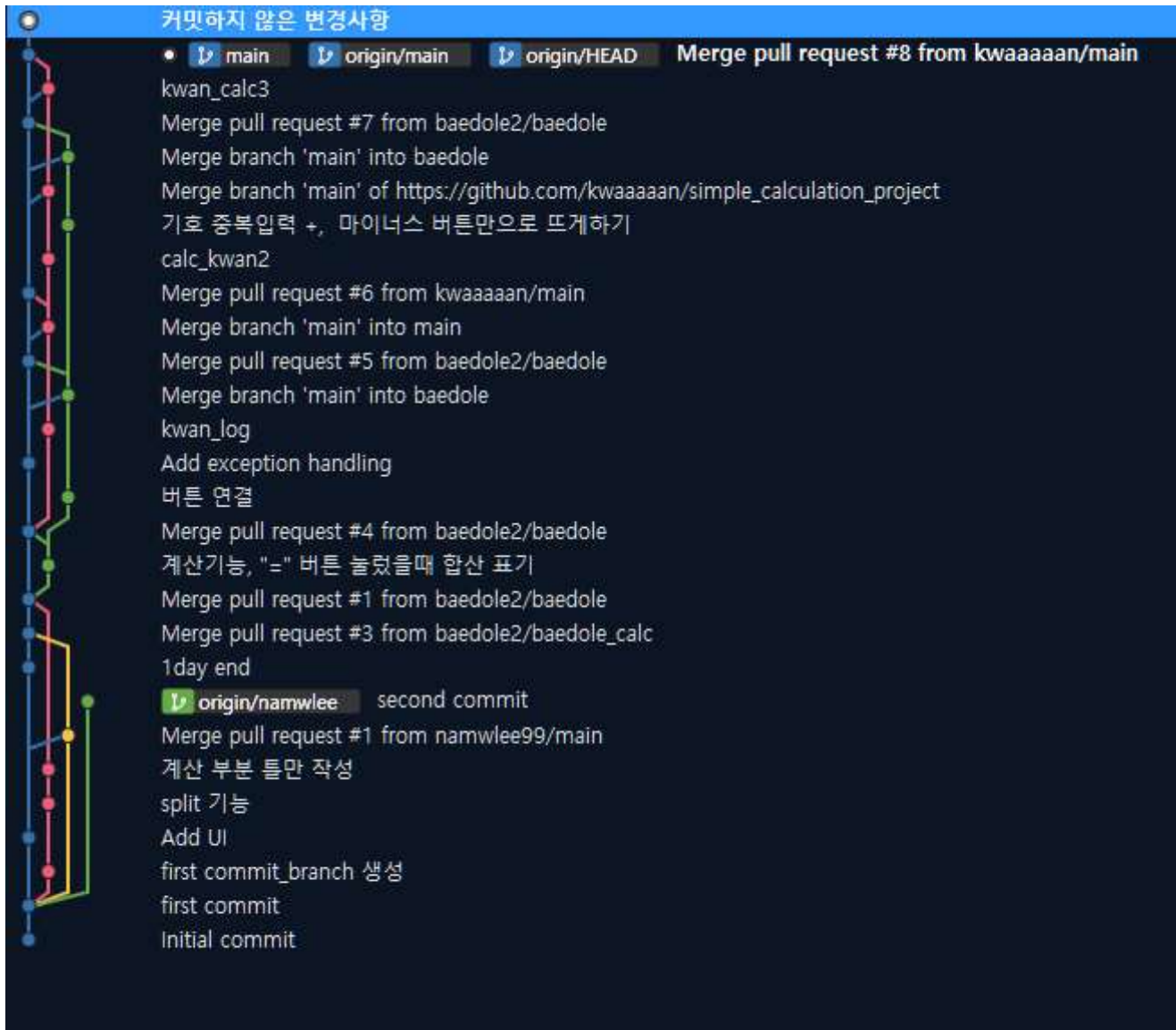
1시간

2. 프로젝트 분석 및 느낌점

1. 결과물



2. git 로그



3. 잘한점

이번 프로젝트를 통해 git을 통해서 branch를 나누어서 프로젝트를 진행할 경우 발생하는 문제 등을 겪을, 이번 프로젝트에서 기능을 하나 추가할 때마다 commit을 하고 팀장 외의 팀원에게는 권한을 부여하지 않고 fork를 통해 merge를 할 수 있게 함. 다양한 git경험을 통해 다른 프로젝트를 진행하게 될 경우 많은 도움이 될 것 같음

4. 아쉬운점

팀장이 보기에는 팀원들이 회의에 적극적으로 참여가 부족한 것 같음. 또, 처음 계산기를 어떠한 방식으로 구현을 할지 구체화가 되지 않아 팀원간의 개발방향이 잠깐 틀어졌던 것 같음.

5. 느낀점

김정관

C#이 익숙하지않아 힘들었지만, 팀원들과 함께 해서 한층 성장하게된 계기가 된거 같음.

배용진

파이썬만큼 제공하는 여러 고급기능 들이 많아 c++ 다룰 때 보다 툴 다루기는 편했다.
계산기라고 알보다가 각종 기능들 세부구현으로 들어갈 수록 점점 구현하기 힘들었음.
후반부 될 수록 c# 에서 폼을 이용해서/혹은 객체 참조를 이용해 쉽게 접근 하는 법을 깨달은게 아쉬움.

이남웅

이번 프로젝트는 팀원과의 소통과 빠른 구현을 목표로 잡았습니다. 회의의 적극성이 약간 부족한 점이 있었지만 회의가 끝나면 개발 시간을 정하고 회의를 주기적으로 해서 최대한으로 소통을 자주하려고 하도록 했고, 구현이 조금 어려운 점은 간단한 방식으로 방향을 제시해서 구현을 빠르게 하고 수정하는 방식을 잡았습니다. 약간 아쉬운 점은 있지만 원하는 결과가 나와서 만족스러웠습니다.



이남웅

hello



이전 포스트

Dash 기본 사용 메뉴얼 - 2

0개의 댓글

댓글을 작성하세요

댓글 작성

