

코드 분석 1

Hyunchan, Park

<http://oslab.jbnu.ac.kr>

Division of Computer Science and Engineering

Jeonbuk National University



강의 일정 (updated)

주차	월 (100분)	수 (50분)
1 (9/1)	Introduction	OSS 역사 1
2	OSS 역사 2 / OSS 개요 1	추석
3	OSS 개요 2 / OSS 라이선스 1	OSS 라이선스 2
4	OSS 활용 방법 / 버전 관리 도구 Git 1	버전 관리 도구 Git 2
5 (9/30)	버전 관리 도구 Git 3	버전 관리 도구 Git 4
6	버전 관리 도구 Git 5	한글날
7	특강 1 (10/14)	코드 분석 1
8 (10/21)	중간고사	개인 프로젝트 1 : 프로젝트 선정 및 계획 마감 (휴강)
9	특강 2 (10/28)	코드 분석 2
10 (11/4)	클라우드를 이용한 개발 환경 구축 1	클라우드를 이용한 개발 환경 구축 2
11	GitHub를 이용한 협업 1	GitHub를 이용한 협업 2
12	개인 프로젝트 2: 대상 프로젝트 코드 분석 결과 발표 및 기능 구현 방향, 커뮤니티 참여 방안 발표 및 피드백	
13 (11/25)	코드 리뷰	CI/CD 구축
14	개별 프로젝트 진행	
15 (12/9)	개인 프로젝트 3: 최종 프로젝트 발표, 기능 구현 소개 및 시연, OSS 커뮤니티 활동 소개	

시작 전에...

- Visual Studio 2019 Community 설치
- 뒤 슬라이드 참고하여 Cloning, Build 수행

1단계: 준비

1. 프로젝트의 개요 파악
2. 실제 SW의 동작 확인
3. 수정할 기능의 구상
4. 빌드 및 실행 환경 구성
5. 빌드 후 정상 동작의 확인

1단계: 준비

1. 프로젝트의 개요 파악

- 개발 및 실행 환경 요구 사항
 - 내가 환경을 구성하기에 적절한 자원(컴퓨팅 환경, 시간, 비용 등)을 갖고 있는가?
- 추가 자료들이 있는가?
 - 구글 등 검색에서 나오는 자료들이 많아 trouble shooting이 용이함
 - Project Community 의 규모가 클수록 자료도 많음

2. 실제 SW의 동작 확인

- 실행 환경을 구성해보지 않고도 미리 수행해볼 수 있다면 매우 좋음
 - 예) 소스 코드가 아니라도 실행 파일이 있거나, 앱이 있거나.
- 수정할 대상이 될 기존 기능의 동작을 확인

1단계: 준비

3. 수정할 기능의 구상

- 어떤 기능을 수정, 추가할 것인가?
 - 실제 사용해보면서 하려고 하는 일의 정확한 요구 사항을 파악해야 함
- 구현이 가능할 것인지 미리 예측을 해봄
 - 겉으로 파악할 수 있는 세부 동작을 추측해가며 기능의 구현 방향을 다양하게 구상해보고, 난이도를 미리 예측해보아야 함

4. 빌드 및 실행 환경 구성

- Build environment 의 구성에 관한 문서를 찾고 그에 따라 환경 구성
 - 뭔가 복잡한데 아무런 문서가 없으면...빠르게 포기하자

5. 빌드 후 정상 동작의 확인

- 실행 환경을 구성하고 수정하고자 하는 기능의 정상 동작을 확인

2단계: 구조 분석 (architecture)

1. Documents 의 확보
 - 전체적인 구조에 대한 document 가 존재해야 함
 - 세부 컴포넌트에 관한
2. 전체적인 구조와 각 컴포넌트의 역할, 인터페이스 파악
 - 구조를 설명하는 main document 전체를 처음 한 번은 정독하자
 - 당장 연관이 없어 보이더라도 소스코드를 볼 때 큰 도움이 됨
3. 수정 기능과 연관있는 세부 컴포넌트의 세부 역할 파악
4. 연관된 컴포넌트들 간의 인터페이스 파악

3단계: 코드 분석

1. Document에 따라 코드 분석의 진입점을 설정

- Top-down: 동작 방식(event driven, procedure 등)에 따라 main() 함수와 같이 뼈대가 되는 코드부터 call chain을 따라가며 수정하고자 하는 코드까지 trace를 진행
- Bottom-up: 수정하고자 하는 코드를 바로 접근하고, 해당 코드를 호출하는 함수들을 추적해나가며 전체 구조를 파악

2. 수정하고자 하는 코드의 동작을 확인

- 우선 코드에 존재하는 주석들을 통해 코드 전체의 흐름 파악
- 간단한 debugging message를 통해 세부적인 사항 파악: 변수 및 sub-function의 기능 확인

3단계: 코드 분석

3. 관련된 코드, 컴포넌트와의 연관 관계를 파악

- 컴포넌트, 함수들 간의 관계를 잘 분리해서 설계하지 않은 코드의 경우, 해당 코드를 수정하면 다른 코드에 영향을 미치는 경우가 있음
- 전체 코드가 공유하는 자료 구조, 전역 변수 등을 확인해야 함

4. 코드 수정

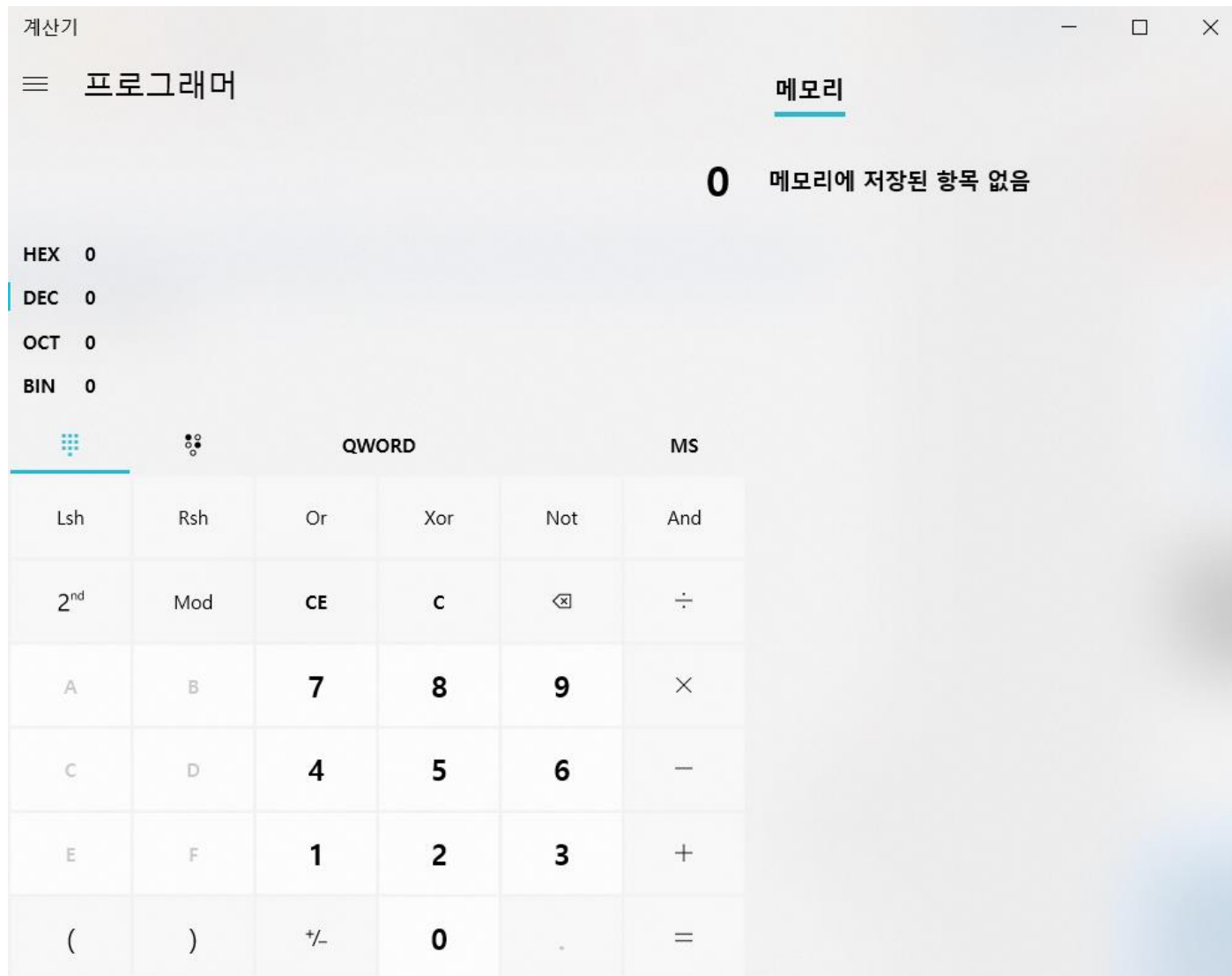
- 수정하고자 하는 기능의 형태, 동작을 설계하고,
- Divide & Conquer 방식으로 진행할 수 있도록 구현의 단계를 세분화
- 한 단계씩 진행하며 반드시 필요한 테스트들을 진행
- 기존 코드 내용을 수정할 경우, 최소한 주석 처리를 해두거나, `#ifdef` 와 같은 전처리 명령을 이용해 수정한 부분을 기존 코드와 분리할 것

윈도우 계산기 앱 수정

<https://github.com/microsoft/calculator>



계산기 앱



계산기 앱

- 프로그래머의 친구, 계산기!
 - Window key + R, Calc
- 2019년 3월 GitHub에 공개되어 활발하게 진행 중
 - 18K Star, 3K Fork: 높은 인기
 - 52명의 Contributor, 그 중 6-8명 정도가 활발히 활동 중
 - 이슈, Pull Request도 상당수
 - Documentation 도 충실하고, PR Guideline 등도 상세히 제시함
 - 모두가 사용하며, 어렵지 않고, 빌드 부담도 적어 접근하기 좋음
 - 모범적으로 운영 중인 프로젝트



첫 번째 시도: 실패

- 더하기를 수행하면 빼기가 수행되게 수정
 - 처음에는 단순히 + 연산자를 사용해 계산하고 결과를 보여주리라 예상
 - 실제 확인해보니 각 자릿수 별로 더하기를 수행하여 최종 결과 도출 (대단히 큰 숫자의 연산을 위해서 이렇게 구현한 듯)
 - 또한 빼기 연산을 위한 코드는 따로 없고, 더하기 코드를 이용하며 단순히 음수 양수 여부 (sign 값) 만 변경함
 - 최종적으로 더하기 연산에 대해 sign 을 바꿔주는 방식으로 구현하려고 했음
- 실패 원인
 - 내부적으로 더하기 연산을 초기화 과정에 이용함
 - 더하기를 빼기로 바꾸었더니 초기화가 완료되지 않음
 - Break point를 넣어 확인해보았으나, 어떤 루틴인지 파악이 어려움
 - 이것을 확인하려면 초기화 과정을 검토해야 하며, 이는 시간이 많이 소요됨
 - 다른 아이디어로 변경!

시작 전에 설정



개발자용

개발자 기능 사용

이러한 설정은 개발의 용도로만 사용할 수 있습니다.

[자세한 정보](#)

☐ Microsoft Store 앱

Microsoft Store의 앱만 설치합니다.

☒ 테스트용으로 앱 로드

회사와 같은 신뢰할 수 있는 다른 원본의 앱을 설치합니다.

☐ 개발자 모드

신뢰할 수 있는 서명된 앱을 모두 설치하고 고급 개발 기능을 사용합니다.

장치 포털 사용

LAN(Local Area Network) 연결을 통해 원격 진단을 켜세요.

☐ 끄

장치 검색



설정



홈

설정 검색



업데이트 및 보안



Windows 업데이트



배달 최적화



Windows 보안



백업



문제 해결

개발자용

개발자 모드



Microsoft Store 앱

Microsoft Store의 앱만 설치합니다.



테스트용으로 앱 로드

회사와 같은 신뢰할 수 있는 다른 원본의 앱을 설치합니다.

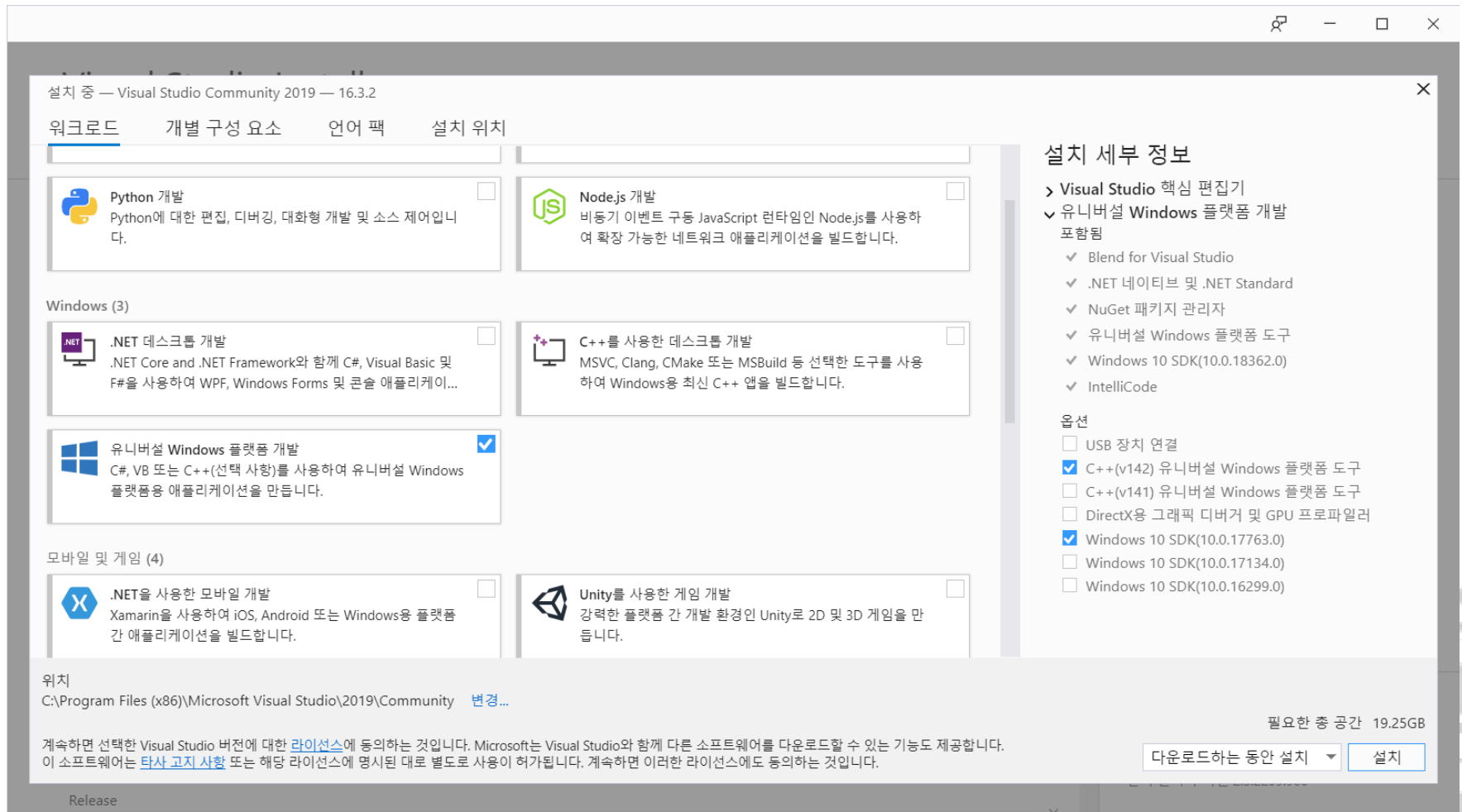


개발자 모드

신뢰할 수 있는 서명된 앱을 모두 설치하고 고급 개발 기능을 사용합니다.

개발자 모드 패키지가 설치되었습니다. 이제 데스크톱용 원격 도구를 사용할 수 있습니다.

빌드 환경 구성 (With VS 2019 Community)





XAML Styler

Team Xavalon | 60,079 installs | ★★★★★ (27) | Free

XAML Styler is a visual studio extension, which formats XAML source code by sorting the attributes based on their importance. This tool can help you/your team maintain a better XAML coding style as well as a much better XAML readability.

사각형 캡처(R)

Download

Overview

Rating & Review

XAML Styler for Visual Studio 2017 and Visual Studio 2019

XAML Styler is a visual studio extension, which formats XAML source code by sorting the attributes based on their importance. This tool can help you/your team maintain a better XAML coding style as well as a much better XAML readability.

Find us on [GitHub](#). Release note for the latest version can be found [here](#). Please report issues [here](#). For more information on various features and script integration, check out the [Wiki](#).

Looking for XAML Styler for older versions?

- [Visual Studio 2013 - 2015](#)
- [Visual Studio 2012](#)

Categories

Tools

Coding

Programming Languages

Tags

Xamarin

add-in

Extension

Forms

uwp

WPF

XAML

Works with

Visual Studio 2017, 2019

Resources

[License](#)

Visual Studio 2019

최근 파일 열기(R)

Visual Studio를 사용할 때 여는 프로젝트, 폴더 또는 파일은 빠른 액세스를 위해 여기에 표시됩니다.

항상 목록의 맨 위에 표시되도록 자주 여는 항목을 고정할 수 있습니다.

시작



코드 복제 또는 체크 아웃(C)

GitHub 또는 Azure DevOps 같은 온라인 리포지토리에서 코드 가져오기



프로젝트 또는 솔루션 열기(P)

로컬 Visual Studio 프로젝트 또는.sln 파일 열기



로컬 폴더 열기(F)

폴더 내에서 탐색 및 코드 편집



새 프로젝트 만들기(N)

시작하려면 코드 스캐폴딩과 함께 프로젝트 템플릿을 선택하세요.

[코드를 사용하지 않고 계속\(W\) →](#)

- ✓ 대상 GitHub 프로젝트를 Fork 하여, my own remote repo 를 생성

Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.

Read the guide

microsoft / calculator

Watch

539

Star

18,724

Fork

3,003

Code

Issues 114

Pull requests 22

Actions

Projects 1

Security

Insights

Windows Calculator: A simple yet powerful calculator that ships with Windows

uwp

windows

windows-10

xaml

c++

csharp

271 commits

8 branches

0 releases


51 contributors

MIT


Learn Git and GitHub without any code!

Using the Hello World guide, you'll start a branch, write comments, and open a pull request.


[Read the guide](#)

 **hyunchan-park / calculator**
forked from [microsoft/calculator](#)

 Watch ▾ 0

 Star 0

 Fork 3,004

 Code

 Pull requests 0

 Projects 0

 Security

 Insights

 Settings

Forking microsoft/calculator

It should only take a few seconds.

 Refresh



✓ 내 remote repo 준비 완료. Local repo 에 clone 해서 작업하고, 여기다 push 할 것

hyunchan-park / calculator
forked from microsoft/calculator

Watch 0

Star 0

Fork 3,004

Code

Pull requests 0

Projects 0

Security

Insights

Settings

Windows Calculator: A simple yet powerful calculator that ships with Windows

Edit

Manage topics

271 commits

8 branches

0 releases

51 contributors

MIT

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

This branch is even with microsoft:master.

joseartrivera Update Calculator keyboard for greater consistency and extensibility (m...)

.github Update bug/feature templates to track if submitter requests

Tools Secondary formatting changes (microsoft#489)

build Move unit test runs to hosted pools (microsoft#671)

docs Always-on-Top mode implemented (microsoft#579)

src Update Calculator keyboard for greater consistency and extensibility (m...

Clone with HTTPS

Use SSH

Use Git or checkout with SVN using the web URL.

https://github.com/hyunchan-park/calcula



Open in Desktop

Download ZIP

22 days ago

2 months ago

5 days ago

코드 복제 또는 체크 아웃

Git 리포지토리 URL 입력

리포지토리 위치(R)

https://github.com/hyunchan-park/calculator.git

로컬 경로(L)

C:\Users\phcph\Documents\GitHub\calcd

...

리포지토리 찾아보기

✓ GitHub Desktop 이 관리하는 주소로 설치하였음 (다른 곳도 무방)

☁ Azure DevOps

🐙 GitHub

뒤로(B)

복제(C)

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+...) Calculator

Debug ARM Calculator(Universal Windows) 원격 컴퓨터

✓ ARM 으로 되어있는 경우, x86 으로 변경

솔루션 탐색기

- 솔루션 탐색기 검색(Ctrl+;)
- 솔루션 'Calculator' (6/6개 프로젝트)
- Solution Items
- CalcManager(Universal Windows)
- Calculator(Universal Windows)**
- CalculatorUITestFramework
- CalculatorUITests
- CalculatorUnitTests(Universal Windows)
- CalcViewModel(Universal Windows)

솔루션 탐색기 팀 탐색기

속성

Calculator 솔루션 속성

기타

(이름)	Calculator
경로	C:\Users\Wphcph\Documents\W
설명	
시작 프로젝트	Calculator
활성 구성	Debug ARM

(이름)
솔루션 파일의 이름입니다.

준비

0 0 calc master 1

Visual Studio 2019 interface showing the Calculator project in the Solution Explorer. The project is a C++ application targeting x86 architecture. The Solution Explorer shows the project structure, including source files and headers. The Output window displays the build output, which is highlighted in red.

Calculator

파일(F) 편집(E) 보기(V) 프로젝트(P) 빌드(B) 디버그(D) 테스트(S) 분석(N) 도구(T) 확장(X) 창(W) 도움말(H) 검색 (Ctrl+...) Calculator

로컬 컴퓨터

솔루션 탐색기

솔루션 탐색기 검색(Ctrl+;)

HistoryItemViewModel.cpp
HistoryItemViewModel.h
HistoryViewModel.cpp
HistoryViewModel.h
MemoryItemViewModel.cpp
MemoryItemViewModel.h
packages.config
pch.cpp
pch.h
StandardCalculatorViewModel.cpp
StandardCalculatorViewModel.h
targetver.h
UnitConverterViewModel.cpp
UnitConverterViewModel.h

속성

HistoryViewModel.h 파일 속성

기타

(이름)	HistoryViewModel.h
내용	False
상대 경로	HistoryViewModel.h
전체 경로	C:\Users\phcph\Documents\GitHub\calc\src\CalculatorUITestFramework\bin\Debug
파일 형식	C++ 헤더 파일

(이름)
파일 개체의 이름을 지정합니다.

출력

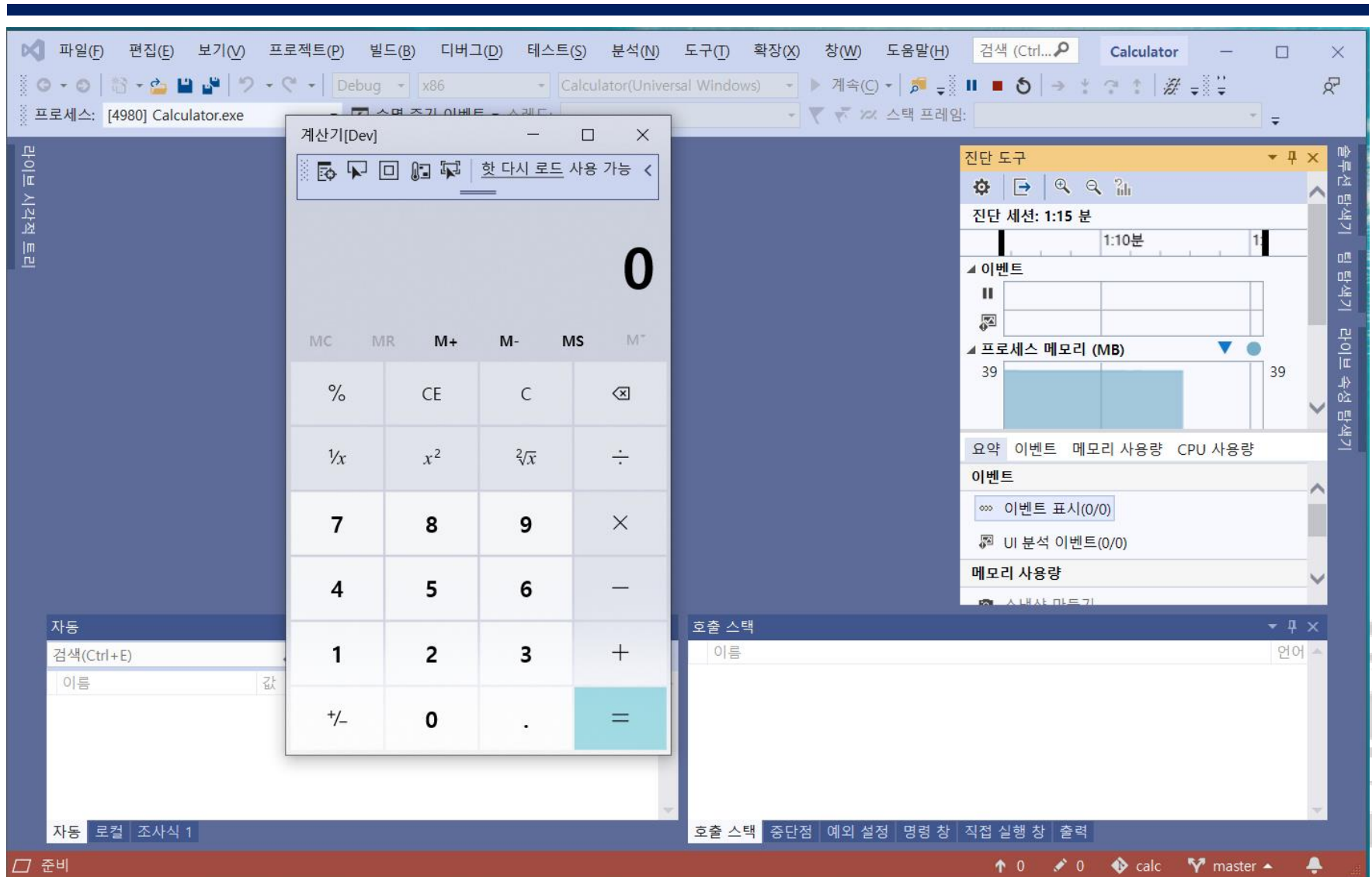
출력 보기 선택(S): 빌드

```
1>C:\Users\phcph\Documents\GitHub\calc\src\CalculatorUITestFramework\CalculatorUITestFramework.csproj : wa
1>CalculatorUITestFramework -> C:\Users\phcph\Documents\GitHub\calc\src\CalculatorUITestFramework\bin\Debu
1>"CalculatorUITestFramework.csproj" 프로젝트를 빌드했습니다.
2>----- 빌드 시작: 프로젝트: CalculatorUITests, 구성: Debug Any CPU -----
2>CalculatorUITests -> C:\Users\phcph\Documents\GitHub\calc\src\CalculatorUITests\bin\Debug\netcoreapp2.1\
===== 빌드: 성공 2, 실패 0, 최신 4, 생략 0 =====
```

오류 목록 출력

준비

빌드 및 수행 확인 (15분 이상 소요됨)



Documentation 확인

ViewModel

The ViewModel layer is contained in the [CalcViewModel](#) project. ViewModels provide a source of data for the UI to bind against and act as the intermediary separating pure business logic from UI components that should not care about the model's implementation. Just as the View layer consists of a hierarchy of XAML files, the ViewModel consists of a hierarchy of ViewModel files. The relationship between XAML and ViewModel files is often 1:1. Here are the notable ViewModel files to start exploring with:

- [ApplicationViewModel.h](#): The ViewModel for [MainPage.xaml](#). This ViewModel is the root of the other mode-specific ViewModels. The application changes between modes by updating the `Mode` property of the `ApplicationViewModel`. The ViewModel will make sure the appropriate ViewModel for the new mode is initialized.
- [StandardCalculatorViewModel.h](#): The ViewModel for [Calculator.xaml](#). This ViewModel exposes functionality for the main three Calculator modes: Standard, Scientific, and Programmer.
- [DateCalculatorViewModel.h](#): The ViewModel for [DateCalculator.xaml](#).
- [UnitConverterViewModel.h](#): The ViewModel for [UnitConverter.xaml](#). This ViewModel implements the logic to support every converter mode, including Currency Converter.

Model

The Model for the Calculator modes is contained in the [CalcManager](#) project. It consists of three layers: a `CalculatorManager`, which relies on a `CalcEngine`, which relies on the `Ratpack`.

CalculatorManager

The `CalculatorManager` contains the logic for managing the overall Calculator app's data such as the History and Memory lists, as well as maintaining the instances of calculator engines used for the various modes. The interface to this layer is defined in [CalculatorManager.h](#).

CalcEngine

The `CalcEngine` contains the logic for interpreting and performing operations according to the commands passed to it. It maintains the current state of calculations and relies on the `RatPack` for performing mathematical operations. The interface to this layer is defined in [CalcEngine.h](#).

RatPack

The `RatPack` (short for Rational Pack) is the core of the Calculator model and contains the logic for performing its mathematical operations (using [infinite precision](#) arithmetic instead of regular floating point arithmetic). The interface to this layer is defined in [ratpak.h](#).

num.cpp ✎ ✕ basex.cpp exp.cpp pch.cpp CalculationResult.cpp ✎ ✕ CCommand.h

CalcManager (전역 범위)

```
1  // Copyright (c) Microsoft Corporation. All rights reserved.  
2  // Licensed under the MIT License.  
3  
4  //-----  
5  // Package Title  ratpak  
6  // File           num.c  
7  // Copyright      (C) 1995-97 Microsoft  
8  // Date          01-16-95  
9  //  
10 //  
11 // Description  
12 //  
13 //     Contains number routines for add, mul, div, rem and other support  
14 //     and longs.  
15 //  
16 // Special Information  
17 //  
18 //  
19 //-----  
20 #include <list>  
21 #include <cstring> // for memmove  
22 #include "ratpak.h"  
23  
24 using namespace std;  
25
```

```

--
26 //-----
27 //
28 //    FUNCTION: addnum
29 //
30 //    ARGUMENTS: pointer to a number a second number, and the
31 //                radix.
32 //
33 //    RETURN: None, changes first pointer.
34 //
35 //    DESCRIPTION: Does the number equivalent of *pa += b.
36 //    Assumes radix is the base of both numbers.
37 //
38 //    ALGORITHM: Adds each digit from least significant to most
39 //    significant.
40 //
41 //
42 //-----

```

```

106 da = (((mexp >= a->exp) && (cdigits + a->exp - c->exp > (c->cdigit - a->cdigit))) ? *pcha++ : 0);
107 // Get digit from b, taking padding into account.
108 db = (((mexp >= b->exp) && (cdigits + b->exp - c->exp > (c->cdigit - b->cdigit))) ? *pchb++ : 0);
109
110 // Handle complementing for a and b digit. Might be a better way, but
111 // haven't found it yet.
112 if (fcompla)
113 {
114     da = (MANTTYPE)(radix)-1 - da;
115 }
116 if (fcomplb)
117 {
118     db = (MANTTYPE)(radix)-1 - db;
119 }
120
121 // Update carry as needed
122 cy = da + db + cy;
123 *pchc++ = (MANTTYPE)(cy / (MANTTYPE)radix);
124 cy /= (MANTTYPE)radix;
125 }
126
127 // Handle carry from last step
128 if (cy && !(fcompla || fcomplb))
129 {
130     *pchc++ = cy;
131     c->cdigit++;
132 }

```

✓ Break point: debug 중 해당 코드에서 수행 중단
 ✓ 각종 메모리 상태를 추적할 수 있음

	Format XAML	Ctrl+K, Ctrl+2
	빠른 작업 및 리팩터링...	Ctrl+.
	정의 피킹	Alt+F12
	정의로 이동(G)	F12
	선언으로 이동(A)	Ctrl+F12
	모든 참조 찾기(A)	Shift+F12
	호출 계층 구조 보기(H)	Ctrl+K, Ctrl+T
	헤더/코드 파일 미리 보기(K)	Ctrl+K, Ctrl+J
	헤더/코드 파일 전환(H)	Ctrl+K, Ctrl+O
	테스트 실행(T)	Ctrl+R, T
	테스트 디버그(D)	Ctrl+R, Ctrl+T
	중단점(B)	
	커서까지 실행(N)	Ctrl+F10

	중단점 삽입(R)
	추적점 삽입(T)

```

119     }
120
121     // Update carry as necessary
122     cy = da + db + cy; 경과 시간 2ms 이하
123     *pchc++ = (MANTTYPE)(cy % (MANTTYPE)radix);
124     cy /= (MANTTYPE)radix;
125 }
126
127 // Handle carry from last sum as extra digit

```

✓ 계산기 실행 전에 이미 중단점에 걸려버림. F5 눌러서 skip

num.cpp baseex.cpp exp.cpp pch.cpp CalculationResult.cpp CCommand.h CalcUtils.cpp calc.cpp CalcEngine.h Command.h

CalcManager (전역 범위) _addnum(PNUMBER * pa, PNUMBER b, uint32_t radix)

```
106 da = (((mexp >= a->exp) && (cdigits + a->exp - c->exp > (c->cdigit - a->cdigit))) ? *pcha++ : 0);
107 // Get digit from b, taking padding into account.
108 db = (((mexp >= b->exp) && (cdigits + b->exp - c->exp > (c->cdigit - b->cdigit))) ? *pchb++ : 0);
109
110 // Handle complementing for a and b digit.
111 // haven't found it yet.
112 if (fcompla)
113 {
114     da = (MANTTYPE)(radix)-1 - da;
115 }
116 if (fcomplb)
117 {
118     db = (MANTTYPE)(radix)-1 - db;
119 }
120
121 // Update carry as necessary
122 cy = da + db + cy;
123 *pchc++ = (MANTTYPE)(cy % (MANTTYPE)radix);
124 cy /= (MANTTYPE)radix;
125 }
126
127 // Handle carry from last sum as extra digit
128 if (cy && !(fcompla || fcomplb))
129 {
130     *pchc++ = cy;
131 }
```

100 % 0 15

자동

검색(Ctrl+E) 검색 심도: 3

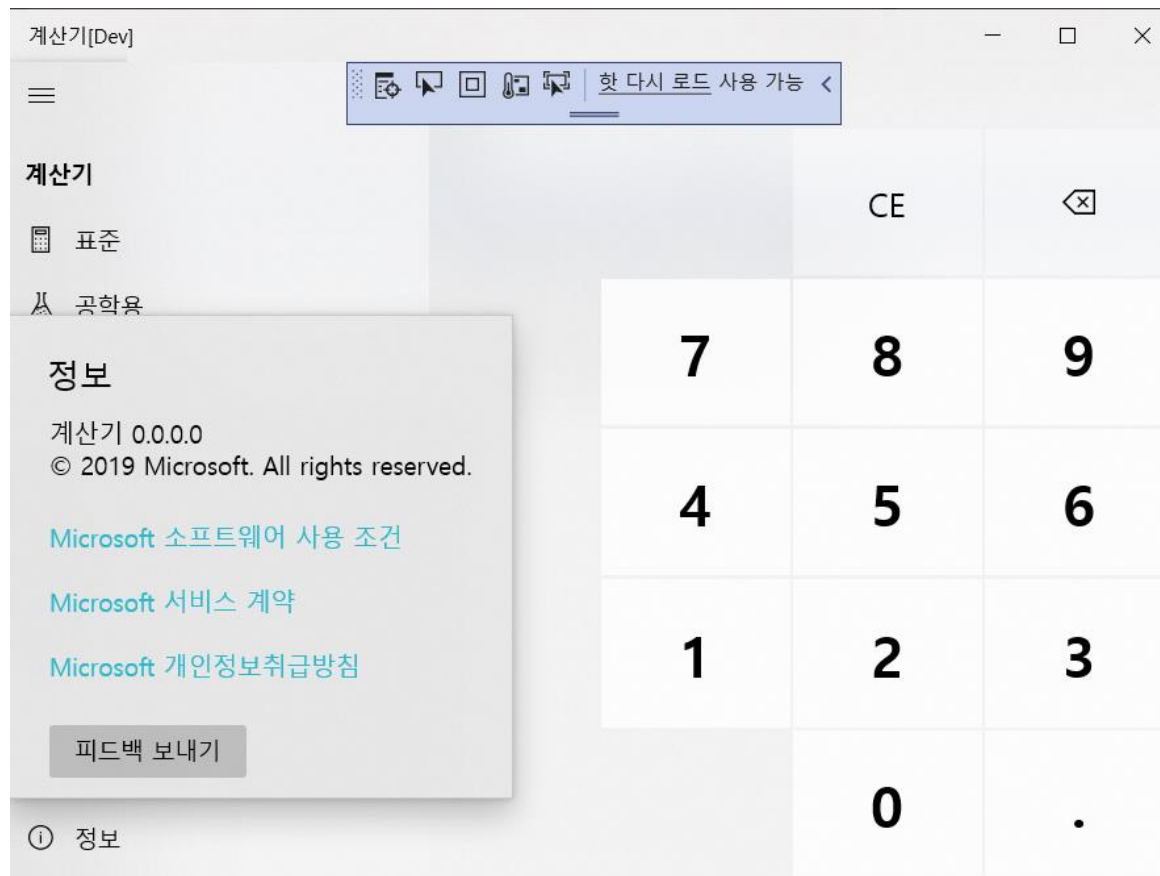
이름	값
cy	0
da	1
db	1

Calculator.exe!CalcEngine::RationalMath::Integer(const CalcEngine...

- ✓ F5 를 여러 번 눌러도 자꾸만 걸리네...
- ✓ 중단점 일단 해제

두 번째 시도

- 정보 창에 내가 원하는 메시지 출력하기
 - 짧은 시간 내에 기능을 수정하기는 쉽지 않아,
좀더 낮은 난이도의 주제를 선택하였음



두 번째 시도

- 텍스트 형태로 존재할 수 밖에 없는 요소가 있으므로,
- 수정해야 하는 코드 부분을 훨씬 쉽게 찾을 수 있음
- 단순 문자열 검색 이용



VS 에서 CTRL+F 문자열 검색

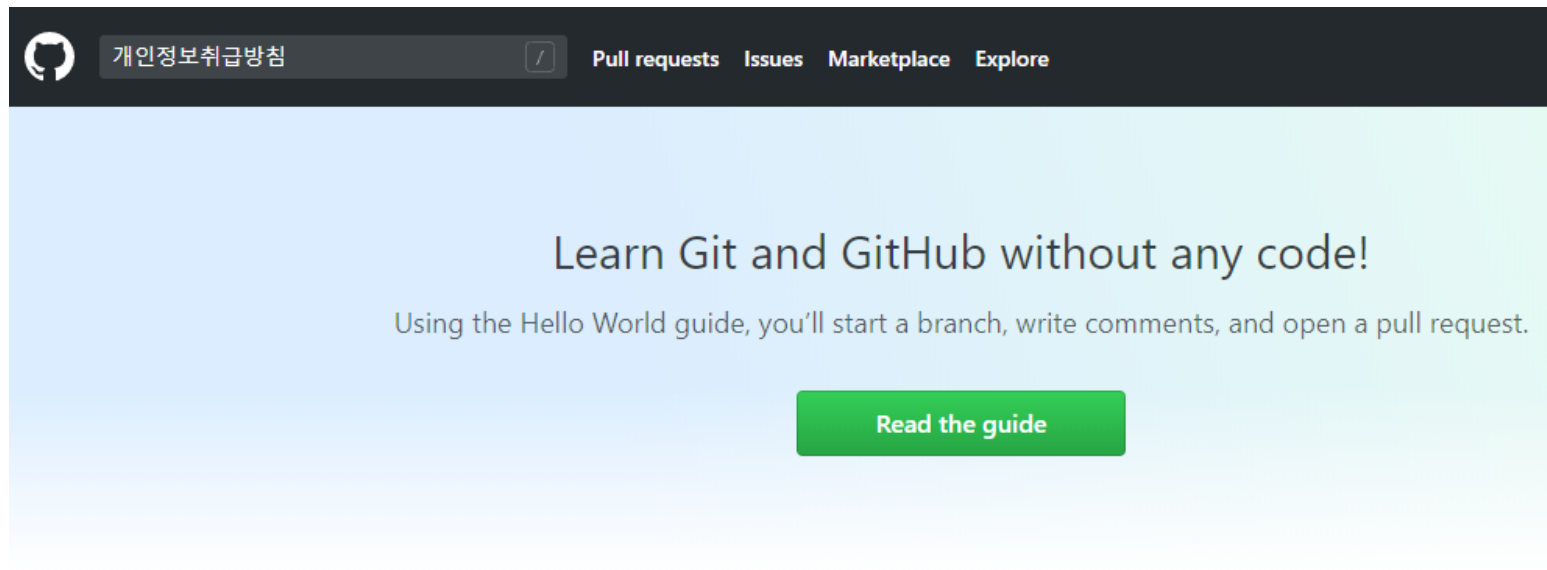
The screenshot displays the Visual Studio search interface. The top panel, titled '찾기 및 바꾸기' (Find and Replace), has two tabs: '파일에서 찾기' (Find in Files) and '파일에서 바꾸기' (Replace in Files). The 'Find in Files' tab is active. The search content is '개인정보취급방침' (Privacy Policy), and the search scope is '전체 솔루션' (All solutions). The '하위 폴더 포함' (Include subfolders) checkbox is checked. Below these are expandable sections for '찾기 옵션' (Find options) and '결과 옵션' (Result options). At the bottom of this panel are buttons for '이전 찾기' (Previous), '다음 찾기' (Next), '모든 책갈피' (All bookmarks), and '모두 찾기' (Find all).

The bottom panel, titled '개인정보취급방침' 찾기 (Find 'Privacy Policy'), shows search results. The toolbar includes a file type filter (set to '모든 파일'), icons for file operations, a group-by dropdown (set to '차례로 경로, 파일'), and a checkbox for '결과 유지' (Keep results). The search criteria are displayed as '모두 찾기 "개인정보취급방침", 하위 폴더, 결과 테이블 찾기, 전체 솔루션, ""'. The results table has columns for '코드' (Code), '파일' (File), '줄' (Line), and '열' (Column).

코드	파일	줄	열
C:\Users\Wphcph\Documents\GitHub\Wcalc\Wsrc\WCalculator\WResources\Wko-kr (1)			
Resources.resw (1)			
<value>Microsoft 개인정보취급방침</value>	Resources.resw	2313	22

At the bottom, a summary line reads: '일치하는 줄 수: 1개 일치하는 파일 수: 1개 검색된 총 파일 수: 424개'.

GitHub 에서도 검색 가능함



The screenshot shows the GitHub homepage. At the top is a dark navigation bar with the GitHub logo, a search bar containing the text '개인정보취급방침', and links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar is a large light blue and green banner with the text 'Learn Git and GitHub without any code!' and 'Using the Hello World guide, you'll start a branch, write comments, and open a pull request.' A green button labeled 'Read the guide' is centered in the banner.

Code	1
Commits	0
Issues	0
Packages	0
Languages	
XML	1

1 code result in [microsoft/calculator](#)

[src/Calculator/Resources/ko-KR/Resources.resw](#)

```
2312 <data name="AboutControlPrivacyStatement.Text" xml:space="preserve">
2313 <value>Microsoft 개인정보취급방침</value>
```

● XML Showing the top match Last indexed on 9 Aug

[Advanced search](#) [Cheat sheet](#)



github.com/microsoft/calculator/blob/24d997dd00534938cb880722108a0f85ae...



```
2304 <data name="AboutControlBackButton" xml:space="preserve">
2305   <value>뒤로</value>
2306   <comment>Content of tooltip being displayed on AboutControlBackButton</comment>
2307 </data>
2308 <data name="AboutFlyoutEULA.Text" xml:space="preserve">
2309   <value>Microsoft 소프트웨어 사용 조건</value>
2310   <comment>Displayed on a link to the Microsoft Software License Terms on the About panel</comment>
2311 </data>
2312 <data name="AboutControlPrivacyStatement.Text" xml:space="preserve">
2313   <value>Microsoft 개인정보취급방침</value>
2314   <comment>Displayed on a link to the Microsoft Privacy Statement on the About panel</comment>
2315 </data>
```

개인정보취급방침

1/1



Tip" xml:s

발견하고, 일단 수정. 그런데 연결 URL 이 없네?

Resources.resw AboutFlyout.xaml

```
2308 <data name="AboutFlyoutEULA.Text" xml:space="preserve">
2309     <value>Microsoft 소프트웨어 사용 조건</value>
2310     <comment>Displayed on a link to the Microsoft Software License Terms on the About panel</comment>
2311 </data>
2312 <data name="AboutControlPrivacyStatement.Text" xml:space="preserve">
2313     <value>운영체제 연구실 링크로 변경함!</value>
2314     <!-- <value>Microsoft 개인정보취급방침</value>!-->
2315     <comment>Edited: Displayed on a link to the Operating Systems Lab. JBNU</comment>
2316     <!-- <comment>Displayed on a link to the Microsoft Privacy Statement on the About panel</comment>
2317     ...
```

Data name 을 검색하여, 참조하는 코드로 이동

The screenshot shows the Visual Studio IDE with the 'AboutFlyout.xaml' file open. The 'Find and Replace' window is active on the left, showing the search term 'AboutControlPrivacyStatement'. The 'XAML' view is selected, and the code for the 'FeedbackButton' is visible. The search results pane at the bottom shows the search results for 'AboutControlPrivacyStatement' in the 'AboutFlyout.xaml' file.

Find and Replace window (찾기 및 바꾸기):

- 찾을 내용(N): AboutControlPrivacyStatement
- 찾는 위치(L): 전체 솔루션
- ☒ 하위 폴더 포함(B)
- 찾기 옵션(O):
- 결과 옵션(S):
- 이전 찾기(P) | 다음 찾기(F) | 모든 책갈피(B) | 모두 찾기(A)

Resources.resw window:

- 13.3" Desktop (1280 x 720) 100% 배율
- 최적: 1280 x 720

XAML view:

```
<Button x:Name="FeedbackButton" x:Uid="FeedbackButton" MinWidth="120" TextWrapping="Wrap"/>
```

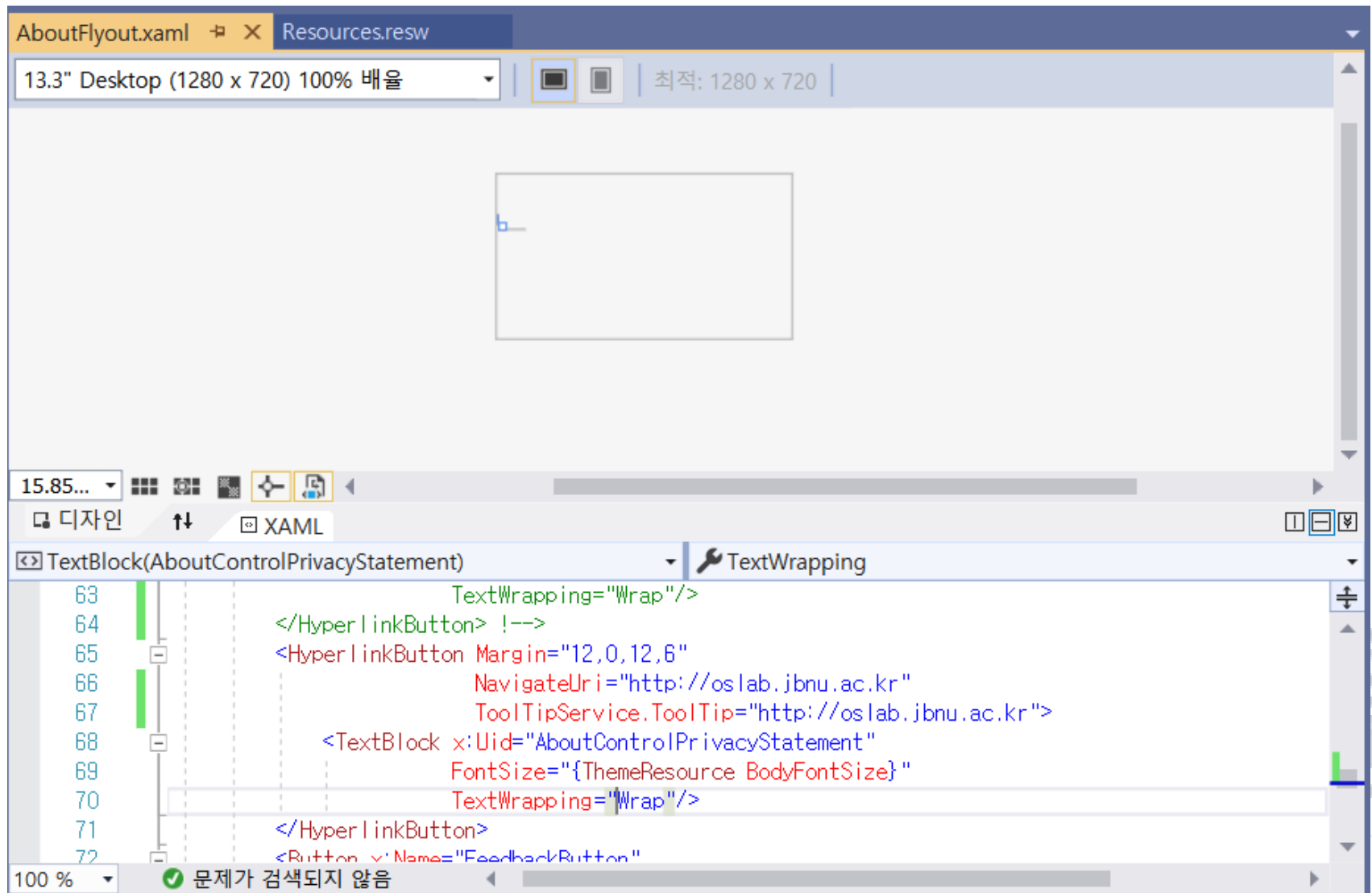
Search Results window ('AboutControlPrivacyStatement' 찾기):

- 모든 파일
- 그룹화 방법: 차례로 경로, 파일
- 결과 유지(K)
- 찾기 결과 검색
- 모두 찾기 "AboutControlPrivacyStatement", 하위 폴더, 결과 테이블 찾기, 전체 솔루션, ""
- 코드
- 파일
- 줄
- 열
- C:\Users\Wphcph\Documents\GitHub\Wcalc\src\WCalculator (66)
 - AboutFlyout.xaml (1)
 - <TextBlock x:Uid="AboutControlPrivacyStatement" AboutFlyout.xaml 61 35
 - Resources (65)

일치하는 줄 수: 66개 일치하는 파일 수: 66개 검색된 총 파일 수: 424개

'AboutControlPrivacyStatement' 찾기 오류 목록 출력

마찬가지로 주식 처리하고, 링크 수정



다시 빌드 후, 확인

계산기[Dev]

☰

☐

✕

☰

☐

✕

계산기

표준

공학용

정보

계산기 0.0.0.0

© 2019 Microsoft. All rights reserved.

[Microsoft 소프트웨어 사용 조건](#)

[Microsoft 서비스 계약](#)

<http://oslab.jbnu.ac.kr>

운영체제 연구실 링크로 변경함!

피드백 보내기

☰

☐

✕

☰

☐

✕

CE

7

8

9

4

5

6

1

2

3

0

.

Operating Systems Laboratory - ✕ +

← → ↻ 🏠 ⓘ 주의 요함 | oslab.jbnu.ac.kr ☆

OPERATING SYSTEMS LABORATORY

Jeonbuk National University


Welcome!

admin - 2018년 6월 28일 - 미분류

반갑습니다!!

전북대학교 컴퓨터공학부

운영체제 연구실 입니다!



코드 분석 개인 과제

- 계산기 앱의 빌드 및 실행
 - Calc1.jpg
- “정보” 항목을 수정한 두 번째 시도를 따라, 본인 학번을 출력하게 변경
 - Calc2. jpg
- 각자 임의의 수정 내용을 설정해 한 가지 수정을 진행해보기
 - 보고서: 수정한 내용 및 진행 과정을 기술하고, 과정 및 결과에 대한 캡처 화면을 적절히 삽입할 것. 실패한 시도가 있다면 그것도 추가할 것.
 - 분량: A4 5장 이내
- 제출 기한:
 - 11/3 (일) 23:59
 - 지각 감점: 5%p / day (3주 내 제출해야 함)

