

FMBA 2018 사전 교육

프로그래밍 개요

KAIST

경영공학과 석박사통합과정 유승현

rambor12@business.kaist.ac.kr

Github Repository

□ 프로그래밍 교육 자료

https://github.com/Seung-hyeon/KAIST_FMBA_Programming_2018

This repository Search Pull requests Issues Marketplace Explore

Seung-hyeon / KAIST_FMBA_Programming_2018 Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

KAIST FMBA Programming Prior-Education Materials Edit

Add topics

5 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Seung-hyeon some files		Latest commit 871ad68 2 minutes ago
Excel VBA	some files	2 minutes ago
Matlab	some files	2 minutes ago
R	some files	2 minutes ago
README.md	some files	2 minutes ago

README.md

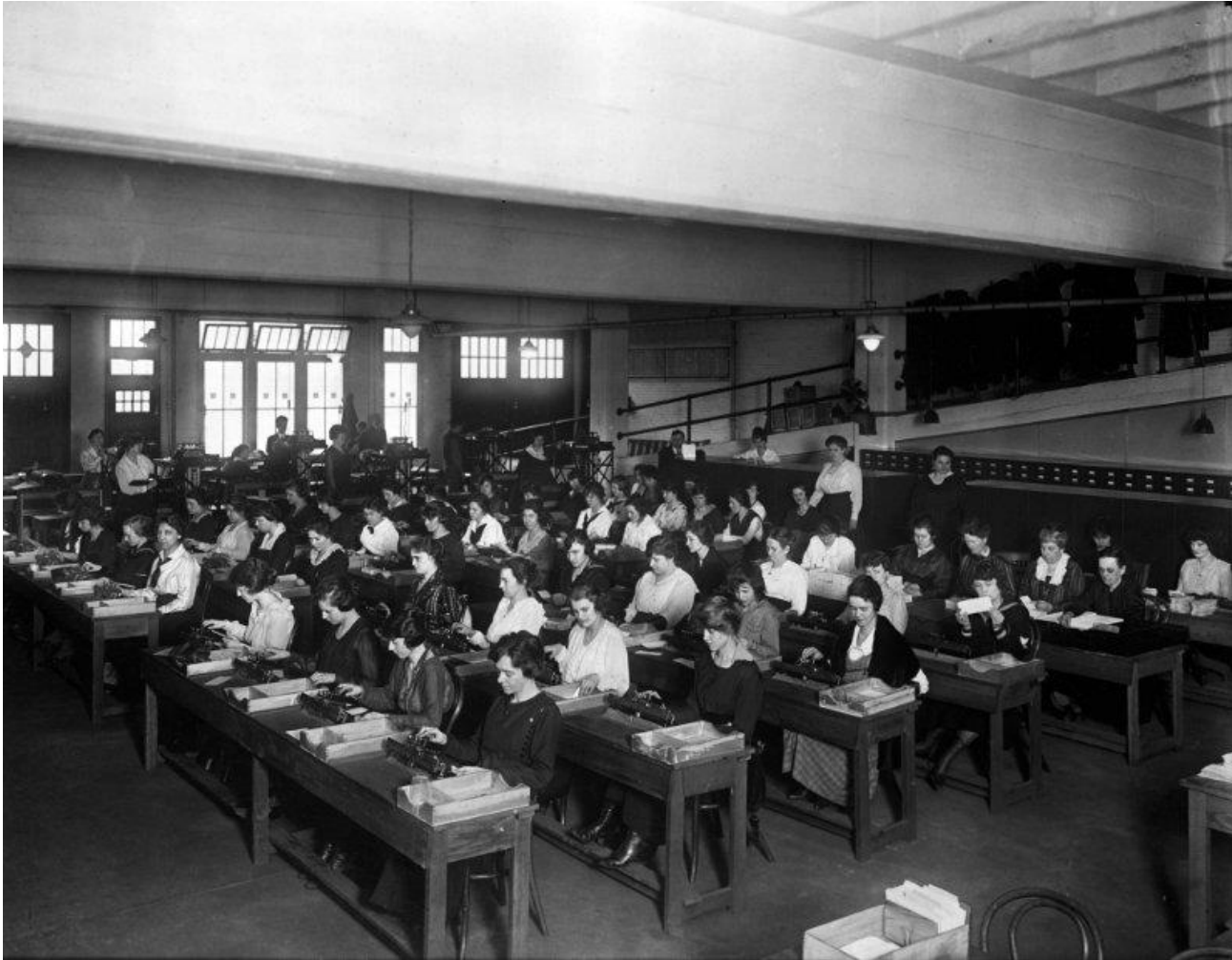
KAIST FMBA Programming 2018

This is KAIST FMBA Programming Prior-Education Materials.

This materials cover Excel VBA , Matlab , and R .

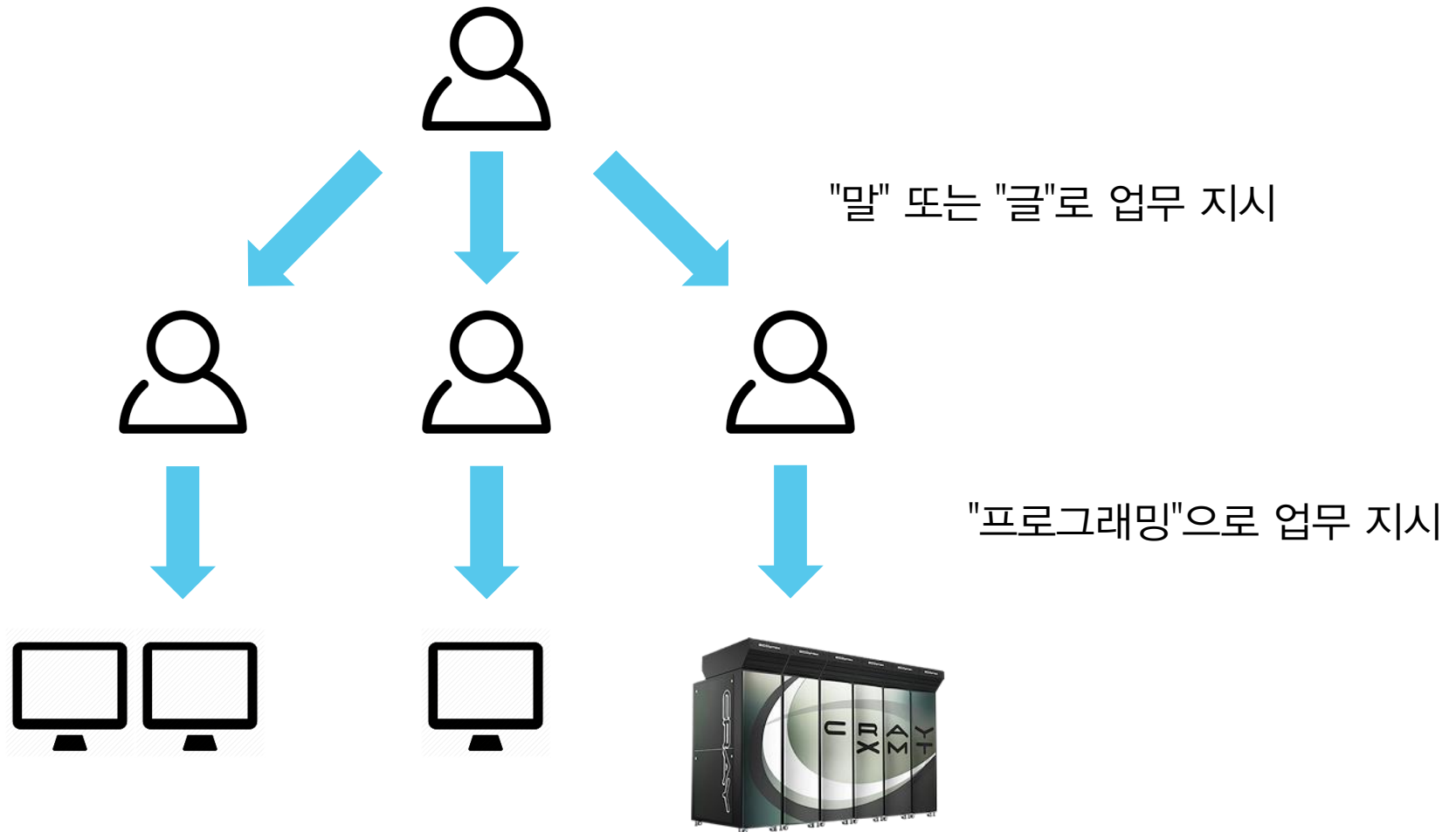
자료 다운로드

컴퓨터



프로그래밍

□ 업무 지시



프로그래밍 사용 예

- 크롤링
- 알고리즘 트레이딩
- 채굴
- 이미지 인식

<https://github.com/tensorflow/models>

1. 금융과 프로그래밍

□ 금융의 특성

- ① 대부분의 자료가 **수치화** 되어 있음.
- ② 이미 **데이터베이스**가 잘 구축 되어 있음.
- ③ 데이터로부터 **새로운 정보**를 찾아내는 것이 중요.
→ 통계, 프로그래밍 기법을 사용

□ 왜 프로그래밍이 필요한가?

- ① 방대한 자료를 **빠르고 효율적으로** 처리.
- ② **실수없이 안정적**으로 일을 처리.
- ③ **어렵고 복잡한** 문제를 쉽게 풀기.

1. 금융과 프로그래밍

□ 금융 데이터베이스

해외 : 블룸버그(Bloomberg), 로이터(Reuter), WRDS, Optionmetric 등

국내 : DataGuide, KIS-Value, QuantWise 등

기타 : 증권사 API, DDE 등

□ 참고 도서



엑셀 VBA 바이블

John Walkenbach 저 | 구미숙, 미현곤 역 | 프리렉 | 2010.11.05.

★★★★★ 10.0 | 네티즌리뷰[2건]

소개 - 독자대상 : **엑셀** 중고급 이상 숙련자 - 구성 : **엑셀** 매크로 기록, 사용자 지향 응용프로그램, 유틸리티 개발 - 특징 : 1) **VBA**를 처음 접하는 독자에게도 이해가 편한 내용과 서술 방식 2) **VBA**를 이해하기 위해 적합한 예제들을 다양하게 수록 3) **엑셀 VBA** 전 영역을 포괄하는 주제들로 구성 4) **엑셀** 2007에...



R언어로 짜는 금융프로그래밍

장기천, 강병진 저 | 서울경제경영 | 2015.12.30.

★★★★★ 0.0 | 네티즌리뷰[0건] | 도서구매 25,000원 → 23,750원(-5%)

소개 ▶ 이 책은 **R언어로 짜는 금융프로그래밍**에 대해 다룬 이론서입니다. R언어 및 금융프로그래밍의 기초적이고 전반적인 내용을 학습할 수 있도록 구성했습니다.

필요한 것만 인터넷 검색하는게 더 유용!
— stackoverflow.com



1. 금융과 프로그래밍

□ 프로그래밍 기본 구조 (Edsger W. Dijkstra's Structured Programming 참조)

1. 변수의 선언 :
(Declaration)

어떤 타입의 데이터를 사용할 것인가?

2. 조건문 :
(Selection)

조건에 따라 어떤 코드를 실행할 것인가?

3. 반복문 :
(Iteration)

이 코드를 언제까지 반복할 것인가?

위의 구조를 이용하여

- 데이터를 어떻게 구조화하고(Data Structure),
 - 어떤 방법을 사용하여(Algorithm),
- 문제를 해결할까?

1. 금융과 프로그래밍

1. 변수의 선언 : (Declaration)

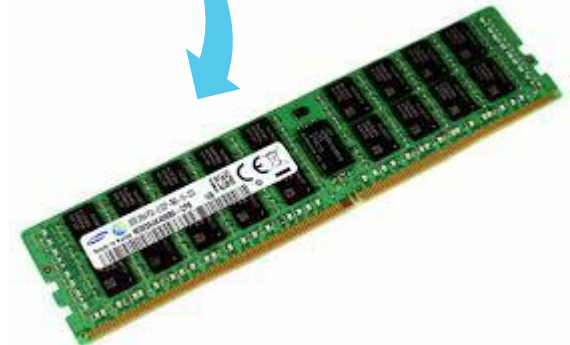
어떤 타입의 데이터를 사용할 것인가?

□ 기초 데이터 타입(Primitive Type)

데이터형	설명	크기(비트)	최소값	최대값
byte	부호있는 정수	8비트	-128	127
int	부호있는 정수	32비트	-2147483648	2147483647
long	부호있는 정수	64비트	-9223372036854775808	9883372036854775807
float	실수	32비트	약 $\pm 1.4 \times 10^{-45}$ (유효숫자 7개)	약 $\pm 3.4 \times 10^{38}$ (유효숫자 7개)
double	실수	64비트	약 $\pm 4.9 \times 10^{-324}$ (유효숫자 15개)	약 $\pm 1.7 \times 10^{308}$ (유효숫자 15개)
char	문자(유니코드)	16비트	'\u0000'(0)	'\uFFFF'(65535)
boolean	True 또는 False	8비트	False	True

변수 할당!

(e.g.) int a = 3



주소(16진수)	값
0xffffffff	
...	
0x80000000	
...	
0x508a0210	3





해당하는 주소 값에 3을 할당

1. 금융과 프로그래밍

1. 변수의 선언 : (Declaration)

어떤 타입의 데이터를 사용할 것인가?

□ 기초 데이터 타입(Primitive Type) 각 언어별 예

데이터형	값	C	 python	 Excel VBA	 MATLAB	
int	3	int a=3;	a=3	Dim a As Integer = 3	a=3;	a=3
long	3	long a=3;	a=3	Dim a As Long = 3	a=3;	a=3
float	3.3	float a=3.3;	a=3.3	Dim a As Single = 3.3	a=3.3;	a=3.3
double	3.3	double a=3.3;	a=3.3	Dim a As Double = 3.3	a=3.3;	a=3.3
char	'A'	char a='A';	a='A'	Dim a As String = 'A'	a='A';	a='A'
boolean	True	사용자가 지정	a=True	Dim a as Boolean = True	a=true;	a=TRUE

1. 금융과 프로그래밍

1. 변수의 선언 : (Declaration)

어떤 타입의 데이터를 사용할 것인가?

□ 참조 데이터 타입(Reference Type)

변수 할당!

(e.g.) `int a[10] = [0,1,2,3,4,5,6,7,8,9]`

데이터형	설명	크기(비트)	최소값	최대값
list or array	배열 혹은 리스트	제한 없음	제한 없음	제한 없음
class	클래스	제한 없음	제한 없음	제한 없음
function	함수	제한 없음	제한 없음	제한 없음



여기서부터 실제 데이터 시작





주소(16진수)	값
0xffffffff	
0x01118708	데이터 시작 주소 값을 할당 참조! 0x508a01b0
...	
0x508a01b0	0
0x508a01d0	1
0x508a01f0	2
0x508a0210	3
0x508a0230	4
0x508a0250	5
0x508a0270	6
0x508a0290	7
0x508a02b0	8
0x508a02d0	9

1. 금융과 프로그래밍

1. 변수의 선언 : (Declaration)

어떤 타입의 데이터를 사용할 것인가?

□ 참조 데이터 타입(Reference Type) 각 언어별 예

데이터형	값	C	 python™	 Excel VBA	 MATLAB	
list or array	[1,2,3]	<code>int a[3]=[1,2,3];</code>	<code>a=[1,2,3]</code>	<code>Dim a(3) As Integer</code> <code>a(1) = 1</code> <code>a(2) = 2</code> <code>a(3) = 3</code>	<code>a=[1,2,3];</code>	<code>a=c(1,2,3)</code>
class				생략		
function	add	<code>int add(int a, int b){</code> <code> return a+b;</code> <code>}</code>	<code>def add(a, b):</code> <code> return a+b</code>	<code>Function add(a, b)</code> <code> add = a+b</code> <code>End Function</code>	<code>function c = add(a, b)</code> <code> c = a+b;</code> <code>end</code>	<code>add <- function(a, b){</code> <code> return(a+b)</code> <code>}</code>

1. 금융과 프로그래밍

1. 변수의 선언 : (Declaration)

어떤 타입의 데이터를 사용할 것인가?

□ 연산자(Operator) : 데이터를 연산하는 함수 예) +, -, ×, ÷,

산술 연산자

비교 연산자

연산자	설명
+	더하기
-	빼기
*	곱하기
/	나누기
^ or **	거듭제곱
==	같은가?
!= or ~=	다른가?
>, <	왼쪽(혹은 오른쪽)이 큰가?
>=, <=	왼쪽(혹은 오른쪽)이 크거나 같은가?

비교 연산자

비교 연산자
(기타)





연산자	설명
& or &&	and, 그리고
or	or, 또는
~ or ^	not, 역
= or <-	값 대입
++	1씩 증가
--	1씩 감소
sizeof or len	데이터 길이

1. 금융과 프로그래밍

2. 조건문 : (Selection)

조건에 따라 어떤 코드를 실행할 것인가?

- 조건의 참, 거짓에 따라 다른 계산을 실행
주로 If ... else 형태의 문법을 사용

C	 python™	 Excel VBA	 MATLAB	
<pre>if(a==3){ printf("a is 3"); }else if(a==3.3{ printf("a is 3.3"); }else{ printf("a is ?"); }</pre>	<pre>if a==3: print('a is 3') elif a==3.3: print('a is 3.3') else: print('a is ?')</pre>	<pre>If a = 3 Then MsgBox "a is 3" Elseif a = 3.3 Then MsgBox "a is 3.3" Else MsgBox "a is ?"</pre>	<pre>if a==3 disp('a is 3') elseif a==3.3 disp('a is 3.3') else disp('a is ?') end</pre>	<pre>if(a==3){ print('a is 3') } else if(a==3.3){ print('a is 3.3') } else { print('a is ?') }</pre>

1. 금융과 프로그래밍





3. 반복문 : (Iteration)

이 코드를 언제까지 반복할 것인가?

□ 같은 명령을 반복해서 실행

반복 횟수가 정해질 경우 : for 루프

반복 횟수가 정해지지 않았을 경우 : while 루프

C	 python™	 Excel VBA	 MATLAB	
<pre>int a = 3; for(i=0; i<=2; i++){ a = a + i; }</pre>	<pre>a = 3 for i in range(3): a = a + i</pre>	<pre>a = 3 For i=0 To 2 a = a + i Next</pre>	<pre>a = 3; for i = 0:2 a = a + i end</pre>	<pre>a = 3 for (i in 0:2){ a = a + i }</pre>

1. 금융과 프로그래밍

- 한가지 프로그래밍 언어만 익숙해지면, 다른 언어는 금세 익힘
- 모든 언어를 다 알아야 할 필요 없음. 필요할 때마다 빨리 배우는게 중요 (배운 후에 금세 까먹어도 괜찮음)
- 지금 배운 VBA, MATLAB, R도 10년 후엔 아무도 안 쓸 수 있음. 그러나 언어의 구조는 크게 바뀌지 않을 것임

추가 : MATLAB 설치

- 다음 실습을 위하여 MATLAB을 미리 설치