

Java를 알고 C배우기

컴퓨터프로그래밍3

week 8-1 다차원 배열-배열 구현

2022.1학기
충남대 조은선

배열의 종류

Static 배열

- ▶ C 전역 배열 변수(`int arr[]={1,2,3};`), Fortran 77 등
- ▶ index의 최대 범위가 compile 할 때 결정됨, 메모리도 compile 할 때 할당

Fixed Dynamic 배열

- ▶ index의 최대 범위가 compile 할 때 결정됨, 메모리는 실행 중 함수 호출될 때 할당
- ▶ Stack에 생성: C 지역 배열 변수 (예: `int arr[10];`)
- ▶ Heap에 생성: Java, C# 배열 (예: `int[] a = new int[5];`) 나중에 다시 나옴

(Non-fixed) Dynamic 배열 나중에 다시 나옴

- ▶ 최대 index 실행시키기 전에는 모름, 메모리도 실행 될 때 할당
- ▶ Stack에 생성: Ada
- ▶ Heap에 생성: Java ArrayList 등 (예: `ArrayList intList = new ArrayList();`)

배열 관리 방법

- ▶ **컴파일**하는 동안 Descriptor 를 가짐
 - ▶ 메모리 할당하는 코드 생성이 목적

원소타입
Index 타입
Index 상한
Index 하한
(시작 주소 관련 정보)

1차원 배열의
컴파일시간 descriptor 예
(C는 파란색 부분만 지원)

원소타입
Index 타입
차원 수
Index 1 상한
Index 1 하한
Index 2 상한
Index 2 하한
(시작 주소 관련 정보)

다차원 배열의
컴파일 시간 descriptor 예
(C는 파란색 부분만 지원)

- ▶ 런타임에도 Descriptor 를 가지면?
 - ▶ 안전 vs. 실행속도
 - ▶ C는 원칙적으로 실행중 Descriptor 를 가지지 않음
 - ▶ Java는 실행 중 Descriptor 를 가짐
`java.lang.ArrayIndexOutOfBoundsException`

다차원 배열의 저장

배열

같은 타입의 여러 요소들을 연속된 메모리에 할당함

Rectangle 배열 (정방형배열)

모든 행이 동일 개수의 요소들을 포함, 모든 열이 동일 개수의 요소들을 포함

(주의: Square 배열이 아님)

저장은?

주로 **행-우선**으로 길게 펼쳐서!

3	5	7
1	0	5
9	3	8

저장 순서: 3 5 7 1 0 5 9 3 8 ..

char a[3][4]; // 선언

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]

a[0][0], a[0][1], a[0][2], a[0][3], a[1][0], a[1][1], a[1][2]...

배열 요소 주소 구하기

a[0][0]	a[0][1]	a[0][2]
a[1][0]	a[1][1]	a[1][2]

1차원 배열 복습

- ▶ `int arr[10];`
- ▶ 0번째 요소의 주소는: `arr` 또는 `&arr[0]`
 - ▶ 요소의 값은 `*arr` 또는 `arr[0]`
- ▶ 1번째 요소의 주소는: `arr+1` 또는 `&arr[1]`
 - ▶ 요소의 값은 `*(arr+1)` 또는 `arr[1]`
- ▶ 2번째 요소의 주소는: `arr+2` 또는 `&arr[2]`
 - ▶ 요소의 값은 `*(arr+2)` 또는 `arr[2]`
- ...
- ▶ i번째 요소의 주소는: `arr+i` 또는 `&arr[i]`
 - ▶ 요소의 값은 `*(arr+i)` 또는 `arr[i]`

2차원 배열의 i,j번째 요소

- ▶ `int arr[2][3];`
- ▶ 0,0번째 : `arr` 또는 `&arr[0][0]`
- ▶ 0,1번째 요소의 주소는: `arr+1` 또는 `&arr[0][1]`
 - ▶ 요소의 값은 `*(arr+1)` 또는 `arr[0][1]`
- ▶ 0,2번째 요소의 주소는: `arr+2` 또는 `&arr[0][2]`
 - ▶ 요소의 값은 `*(arr+2)` 또는 `arr[0][2]`
- ▶ 1,0번째 요소의 주소는: `arr+3` 또는 `&arr[1][0]`
 - ▶ 요소의 값은 `*(arr+3)` 또는 `arr[1][0]`

...

- ▶ i,j번째 요소의 주소 식은 `(arr + i*3+j)` 또는 `&arr[i][j]`
 - ▶ 참고: 주소 계산을 위한 산술연산 `arr + (i*3+j) * sizeof(int)`
 - ▶ 요소의 값은 `*(arr + i*3+j)` 또는 `arr[i][j]`

Quiz

다음 배열의 시작지점이 1024라면,

```
int arr[3][4];
```

아래와 같은 코드에서 100이 지정되는 곳의 주소는?

```
arr[2][1] = 100;
```

(식만 적어도 됩니다)