List vs array vs arrayList

;모두 데이터를 저장하는데 사용되는 컬렉션 타입이다.

1. List

동적으로 크기가 조정될 수 있는 컬렉션

요소의 추가, 삭제, 수정 등이 자유로움

인덱스를 사용하여 요소에 직접 액세스 가능

Generic으로 구현되어 타입 안정성 보장

(\*Generic(;일반화) 다른 데이터 형식을 추가할 수 없도록 안정성을 적용시켜 컬렉션에 저장하고 사용하는 클래스 ; 성능이 향상되고 안정성이 높다는 장점이 있다.

\*배열과 제네릭 리스트의 차이점

:배열은 인덱스나 변수, 혹은 변수 값을 추가하고 싶을 때 프로그램을 수정해야 한다.

하지만 리스트는 추가하고 싶은 값만 추가하면 된다.(간편)

배열은 값이나 길이를 정해놓고 시작하기 때문에 고정되지만, 리스트는 유동적으로 변화를 줄 수 있다.

1. Array

고정된 크기의 요소들을 저장할 수 있는 컬렉션

배열의 크기는 생성 시점에 결정, 변경 불가능

인덱스를 사용하여 요소에 직접 액세스 가능

다차원 배열도 지원

1. ArrayList

동적으로 크기가 조정될 수 있는 컬렉션

요소의 추가,삭제,수정 등이 자유로움

인덱스를 사용하여 요소에 직접 엑세스 가능

박싱,언박싱으로 인한 성능 저하가 있을 수 있음

(\*박싱; 스택에서 힙으로 메모리가 옮겨 가는 것

값 타입을 참조 타입으로 변환하는 프로세스

값 형식의 데이터는 메모리의 스택 영역에 할당되는데 참조타입으로 변환되면 힙 영역에 복사된다.

\*언박싱; 힙에서 스택으로 메모리가 옮겨가는 것

참조 타입을 값 타입으로 변환하는 프로세스

;메모리는 메모리주소 와 값이 하나의 세트로 존재한다.

\*스택; 스택 메모리도 주소와 값이 한 세트로 존재함

\*힙; 힙 메모리도 주소와 값이 한 세트로 존재함

참조 형식이 저장됨

메모리 제한이 없음

스택과 같은 점은 무엇을 저장할지 미리 정해 놓음,

관리 하는 방식이 같음

;데이터 복구가 가능한 이유중 하나가 힙 메모리, 정보를 삭제할 때 그 정보가 힙에 저장되어 있으면, 컴퓨터는 스택 메모리 안 해당 주소값만 삭제함. 사용자 입장에서 주소가 사라져 접근이 불가하지만 실제 정보는 힙 안에 살아있는 것.)



ㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡㅡ

참조

1. <https://ctrlvgames.tistory.com/24>
2. <https://blog.naver.com/sdsss031/222999414923>
3. https://glassnabi.tistory.com/8