

IT@COOKBOOK



그림으로 설명하고 실생활 예제로 실습하는

자바를 부탁해

① 자바 기초

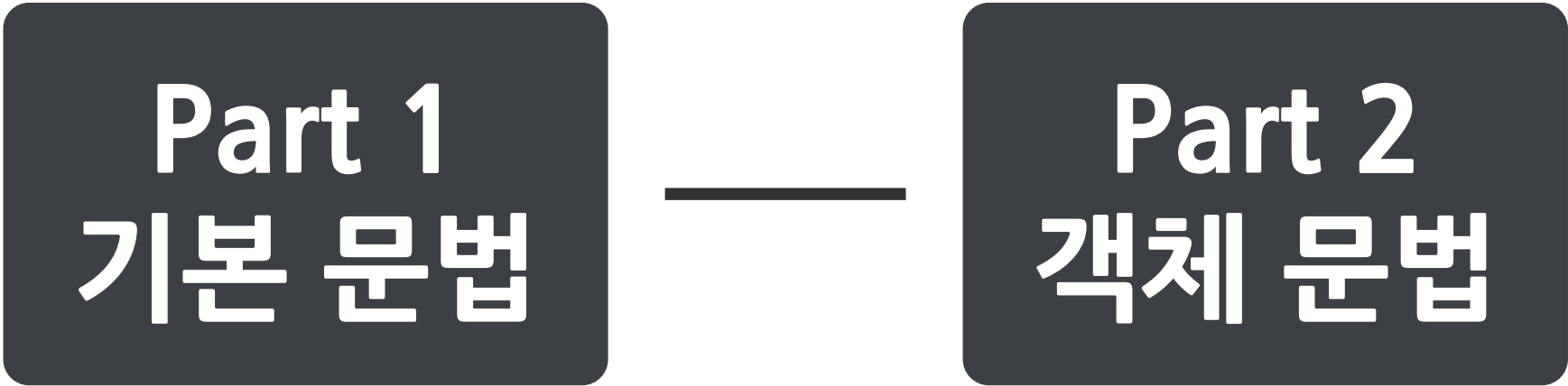
- 01 자바 시작하기
- 02 변수, 타입, 연산자

② 메소드와 조건문

- 03 메소드
- 04 조건문

③ 반복문과 배열

- 05 반복문
- 06 배열



④ OOP 기초

- 07 객체 지향 프로그래밍
- 08 객체 지향 프로그래밍의 활용
- 09 자바 API

⑤ OOP 심화

- 10 상속
- 11 인터페이스
- 12 예외 처리와 파일 입출력

⑥ 프로젝트

- 13 실전 프로젝트: 영화 예매
관리 프로그램

CH09

자바 API

01 자바 API 개요

02 자바 API 기초 활용

자바 API의 개념을 이해한다.

다양한 자바 API 사용법을 훈련한다.

CH09

자바 API

01 자바 API 개요

02 자바 API 기초 활용

I . 자바 API란

자바 API(application programming interface)란

- 프로그램 개발을 위한 도구로, 자바가 제공하는 코드

자바 API의 대표적 예로

- System.out.printf(), Math.random(), Math.PI 등이 있음

코드가 미리 준비되어 있어 더 편리한 프로그램을 작성을 도움

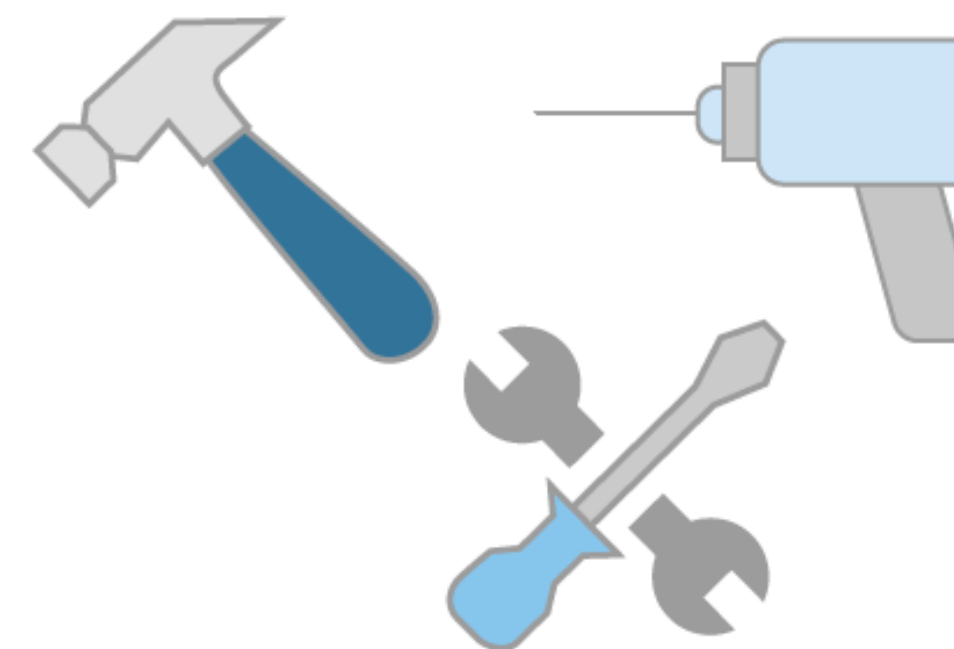
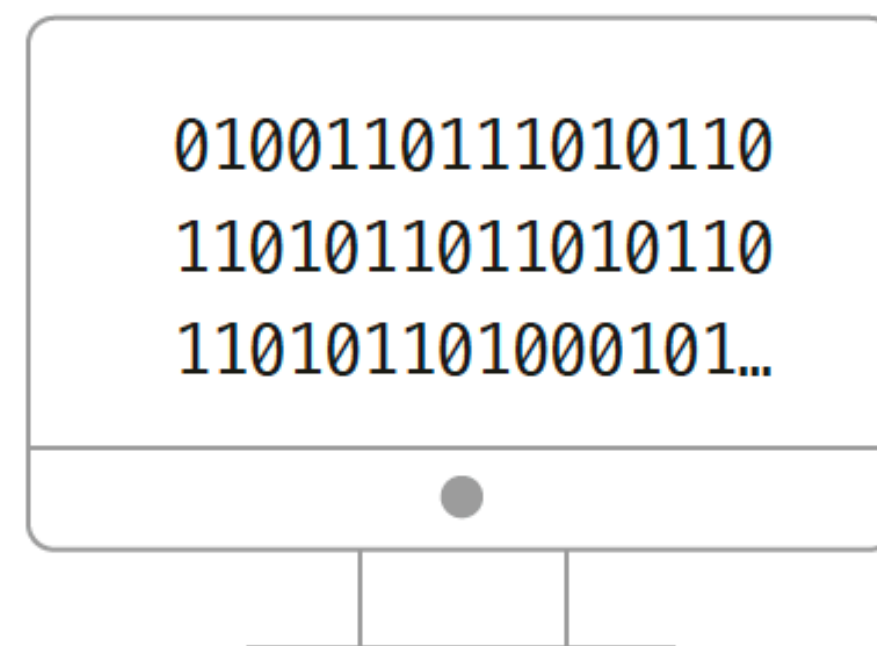


그림 9-1 자바 API = 생활 속 여러 도구들

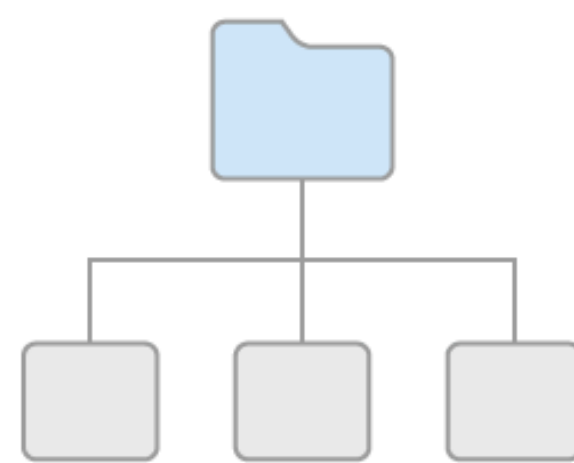
II. 패키지

자바 API를 활용하기 위해서는 먼저 패키지에 대해 알아야 함

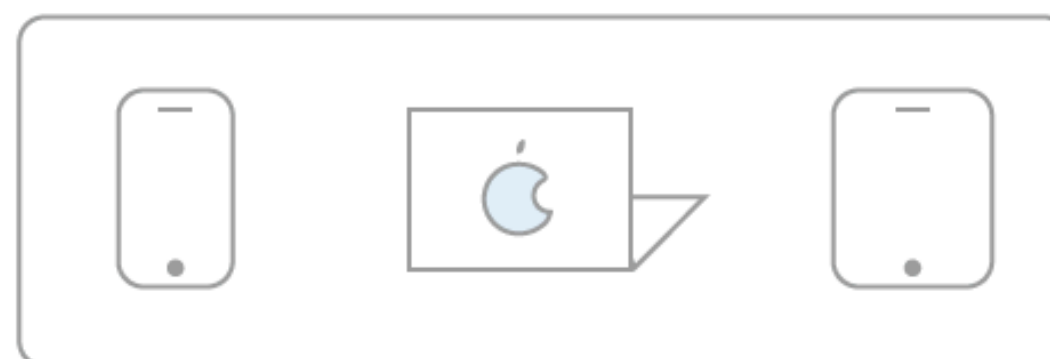
- 자바 API는 일반적으로 패키지에 담겨 제공되기 때문

패키지(package)란 자바 코드의 묶음으로, 일종의 **폴더**(디렉터리)

- 관련 코드를 한곳에 모아 관리하거나
- 이름만 같은 다른 코드를 구분하기 위해 사용

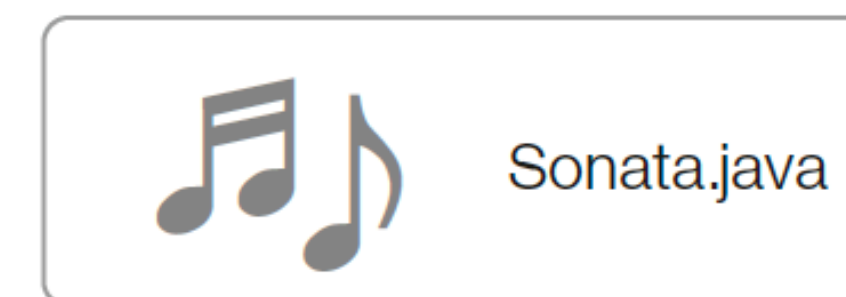


`package apple;`



iPhone.java Macbook.java iPad.java

`package music;`



`package car;`



그림 9-2 패키지 예

III. 자바 API 사용 예

키보드 입력을 받아오는 자바 API, Scanner 클래스를 사용해보자!

불러오기

- Scanner 클래스를 사용하려면 먼저 코드를 불러와야 함
- 사용할 API의 정보를 **import** 키워드와 함께 입력

```
import java.util.Scanner;
```

패키지 클래스

사용하기

- 다음 실습 코드를 통해 확인!

실습 9-1Scanner 클래스: 사용자 입력받기

ch09/ex9_1/UserInput.java

01import java.util.Scanner;-----API 불러오기

02

03public class UserInput {

04 public static void main(String[] args) {

05 Scanner input = new Scanner(System.in);-----스캐너 객체 생성

06 System.out.print("이름: ");

07 String name = input.next();-----문자열 입력받기

08 System.out.print("학번: ");

09 int number = input.nextInt();-----정수 입력받기

10 System.out.print("학점: ");

11 double grade = input.nextDouble();-----실수 입력받기

12 System.out.printf("[%s]님의 학번은 %d이며, 학점은 %.2f 입니다.",

13 name, number, grade);

14 }

15 }

실행 결과

이름: 홍팩

학번: 2021122104-----키보드 입력값

학점: 3.81

[홍팩]님의 학번은 2021122104이며, 학점은 3.81 입니다.

실습 코드

https://github.com/cloudstudying-kr/JavaPlz/blob/master/src/part2/ch09/ex9_1/UserInput.java

실습 9-1

Scanner 클래스: 사용자 입력받기

ch09/ex9_1/UserInput.java

01 import java.util.Scanner;

02

03 public class UserInput {

04 public static void main(String[] args) {

05 Scanner input = new Scanner(System.in);

06 System.out.print("이름: ");

07 String name = input.next();

08 System.out.print("학번: ");

09 int number = input.nextInt();

10 System.out.print("학점: ");

11 double grade = input.nextDouble();

12 System.out.printf("[%s]님의 학번은 %d이며, 학점은 %.2f 입니다.",

13 name, number, grade);

14 }

15 }

API 불러오기

키보드 입력을 읽기 위해 java.util 패키지의 Scanner 클래스를 불러옴

실행 결과

이름: 홍팩

학번: 2021122104

학점: 3.81

[홍팩]님의 학번은 2021122104이며, 학점은 3.81 입니다.

실습 9-1

Scanner 클래스: 사용자 입력받기

ch09/ex9_1/UserInput.java

```
01 import java.util.Scanner;-----API 불러오기
02
03 public class UserInput {
04     public static void main(String[] args) {
05         Scanner input = new Scanner(System.in);-----Scanner 객체 생성
06         System.out.print("이름: ");
07         String name = input.next();-----문자열 입력받기
08         System.out.print("학번: ");
09         int number = input.nextInt();-----정수 입력받기
10         System.out.print("학점: ");
11         double grade = input.nextDouble();-----실수 입력받기
12         System.out.printf("[%s]님의 학번은 %d이며, 학점은 %.2f 입니다.",
13             name, number, grade);
14     }
15 }
```

실행 결과

이름: 홍팍

학번: 2021122104-----키보드 입력값

학점: 3.81

[홍팍]님의 학번은 2021122104이며, 학점은 3.81 입니다.

실습 9-1

Scanner 클래스: 사용자 입력받기

ch09/ex9_1/UserInput.java

```
01 import java.util.Scanner;-----API 불러오기
02
03 public class UserInput {
04     public static void main(String[] args) {
05         Scanner input = new Scanner(System.in);-----스캐너 객체 생성
06         System.out.print("이름: ");
07         String name = input.next();-----키보드로부터 문자열을 입력받음
08         System.out.print("학번: ");
09         int number = input.nextInt();-----정수 입력받기
10         System.out.print("학점: ");
11         double grade = input.nextDouble();-----실수 입력받기
12         System.out.printf("[%s]님의 학번은 %d이며, 학점은 %.2f 입니다.",
13             name, number, grade);
14     }
15 }
```

실행 결과

이름:

학번: -----키보드 입력값

학점:

[홍박]님의 학번은 2021122104이며, 학점은 3.81 입니다.

실습 9-1

Scanner 클래스: 사용자 입력받기

ch09/ex9_1/UserInput.java

```
01 import java.util.Scanner; ..... API 불러오기
02
03 public class UserInput {
04     public static void main(String[] args) {
05         Scanner input = new Scanner(System.in); ..... 스캐너 객체 생성
06         System.out.print("이름: ");
07         String name = input.next(); ..... 문자열 입력받기
08         System.out.print("학번: ");
09         int number = input.nextInt(); ..... 키보드로부터 정수를 입력받음
10         System.out.print("학점: ");
11         double grade = input.nextDouble(); ..... 실수 입력받기
12         System.out.printf("[%s]님의 학번은 %d이며, 학점은 %.2f 입니다.",
13             name, number, grade);
14     }
15 }
```

실행 결과

```
이름: 홍팍
학번: 2021122104 ..... 키보드 입력값
학점: 3.81
[홍팍]님의 학번은 2021122104이며, 학점은 3.81 입니다.
```


실습 9-1

Scanner 클래스: 사용자 입력받기

ch09/ex9_1/UserInput.java

```

01 import java.util.Scanner; ..... API 불러오기
02
03 public class UserInput {
04     public static void main(String[] args) {
05         Scanner input = new Scanner(System.in); ..... 스캐너 객체 생성
06         System.out.print("이름: ");
07         String name = input.next(); ..... 문자열 입력받기
08         System.out.print("학번: ");
09         int number = input.nextInt(); ..... 정수 입력받기
10         System.out.print("학점: ");
11         double grade = input.nextDouble(); ..... 키보드로부터 실수를 입력받음
12         System.out.printf("[%s]님의 학번은 %d이며, 학점은 %.2f 입니다.",
13             name, number, grade);
14     }
15 }

```

실행 결과

```

이름: 홍팩
학번: 2021122104 ..... 키보드 입력값
학점: 3.81
[홍팩]님의 학번은 2021122104이며, 학점은 3.81 입니다.

```

실습 9-1

Scanner 클래스: 사용자 입력받기

ch09/ex9_1/UserInput.java

```
01 import java.util.Scanner; ..... API 불러오기
02
03 public class UserInput {
04     public static void main(String[] args) {
05         Scanner input = new Scanner(System.in); ..... 스캐너 객체 생성
06         System.out.print("이름: ");
07         String name = input.next(); ..... 문자열 입력받기
08         System.out.print("학번: ");
09         int number = input.nextInt(); ..... 정수 입력받기
10         System.out.print("학점: ");
11         double grade = input.nextDouble(); ..... 실수 입력받기
12         System.out.printf("[%s]님의 학번은 %d이며, 학점은 %.2f 입니다.",
13             name, number, grade); ..... 최종 결과 출력
14     }
15 }
```

실행 결과

```
이름: 홍팩
학번: 2021122104 ..... 키보드 입력값
학점: 3.81
[홍팩]님의 학번은 2021122104이며, 학점은 3.81 입니다.
```

실습 9-1Scanner 클래스: 사용자 입력받기

ch09/ex9_1/UserInput.java

```
01 import java.util.Scanner;-----API 불러오기
02
03 public class UserInput {
04     public static void main(String[] args) {
05         Scanner input = new Scanner(System.in);-----스캐너 객체 생성
06         System.out.print("이름: ");
```

Scanner 객체를 활용하여 키보드 입력 문자열을 다양한 형태로 반환할 수 있음

```
07         String name = input.next();           // 입력 문자열을 반환
...
09         int number = input.nextInt();          // 입력 문자열을 정수로 반환
...
11         double grade = input.nextDouble();    // 입력 문자열을 실수로 반환
```

```
이름: 홍익
학번: 2021122104-----키보드 입력값
학점: 3.81
[홍팩]님의 학번은 2021122104이며, 학점은 3.81 입니다.
```


IV. 자바 API 문서

자바 API 문서(java API documentation)란

- 자바에서 제공하는 여러 API를 정리한 것으로, 다양한 버전이 있음

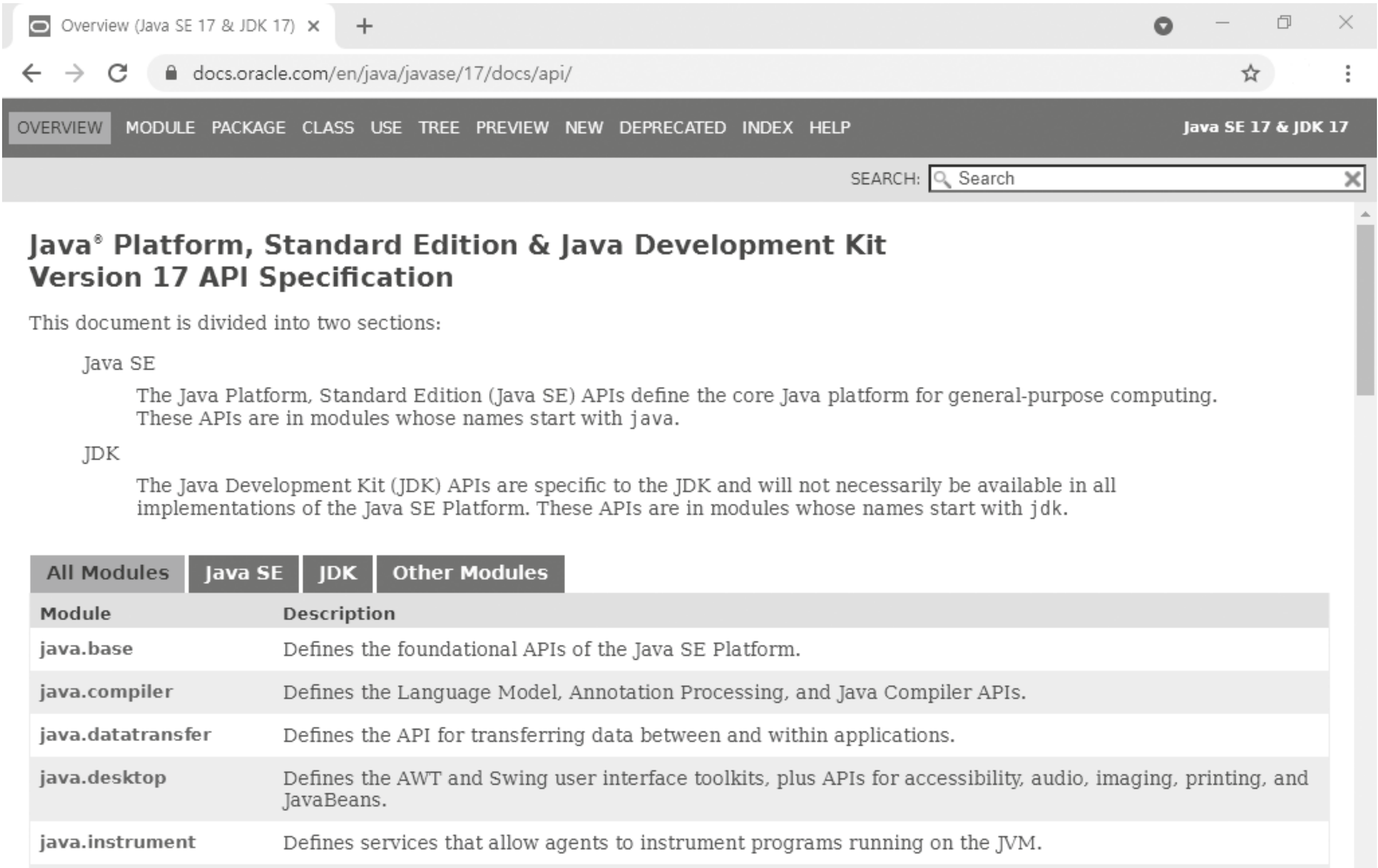


그림 9-3 자바 17 버전 API 문서
<https://docs.oracle.com/en/java/javase/17/docs/api/>

IV. 자바 API 문서

자바 API 문서에서 Scanner를 검색하면

- Scanner 클래스의 개요와 필드, 메소드, 생성자 등의 사용법을 볼 수 있음

Module java.base

Package java.util

Class Scanner

java.lang.Object

java.util.Scanner

All Implemented Interfaces:

Closeable, AutoCloseable, Iterator<String>

public final class Scanner

extends Object

implements Iterator<String>, Closeable

A simple text scanner which can parse primitive types and strings using regular expressions.

A Scanner breaks its input into tokens using a delimiter pattern, which by default matches whitespace. The resulting tokens may then be converted into values of different types using the various next methods.

For example, this code allows a user to read a number from System.in:

Scanner sc = new Scanner(System.in);

int i = sc.nextInt();

그림 9-4 자바 API 문서의 Scanner 클래스 부분 (<https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Scanner.html>)

IV. 자바 API 문서

자바 API 문서에서 Scanner를 검색하면

- Scanner 클래스의 개요와 필드, 메소드, 생성자 등의 사용법을 볼 수 있음

Module java.base
Package java.util
Class Scanner

java.lang.Object
 java.util.Scanner

All Implemented Interfaces:
Closeable, AutoCloseable, Iterator<String>

public final class **Scanner**
extends Object
implements Iterator<String>, Closeable

A simple text scanner which can parse primitive types and strings using regular expressions.

A Scanner breaks its input into tokens using a delimiter pattern, which by default matches whitespace. The resulting tokens may then be converted into values of different types using the various next methods.

For example, this code allows a user to read a number from System.in:

```
Scanner sc = new Scanner(System.in);  
int i = sc.nextInt();
```

 **TIP** 구글링을 통한 학습과 stackoverflow.com 등의 개발자 커뮤니티도 활용하는 것을 추천!

그림 9-4 자바 API 문서의 Scanner 클래스 부분 (<https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Scanner.html>)

Quiz

01 Scanner 클래스의 두 메소드 `next()`와 `nextLine()`의 차이점을 설명하시오.

CH09

자바 API

01 자바 API 개요

02 자바 API 기초 활용

I . Math 클래스

Math 클래스는 수학과 관련된 자바 API

- 지수와 로그, 제곱근과 삼각함수 등의 연산을 제공
- 해당 클래스는 java.lang 패키지에 존재

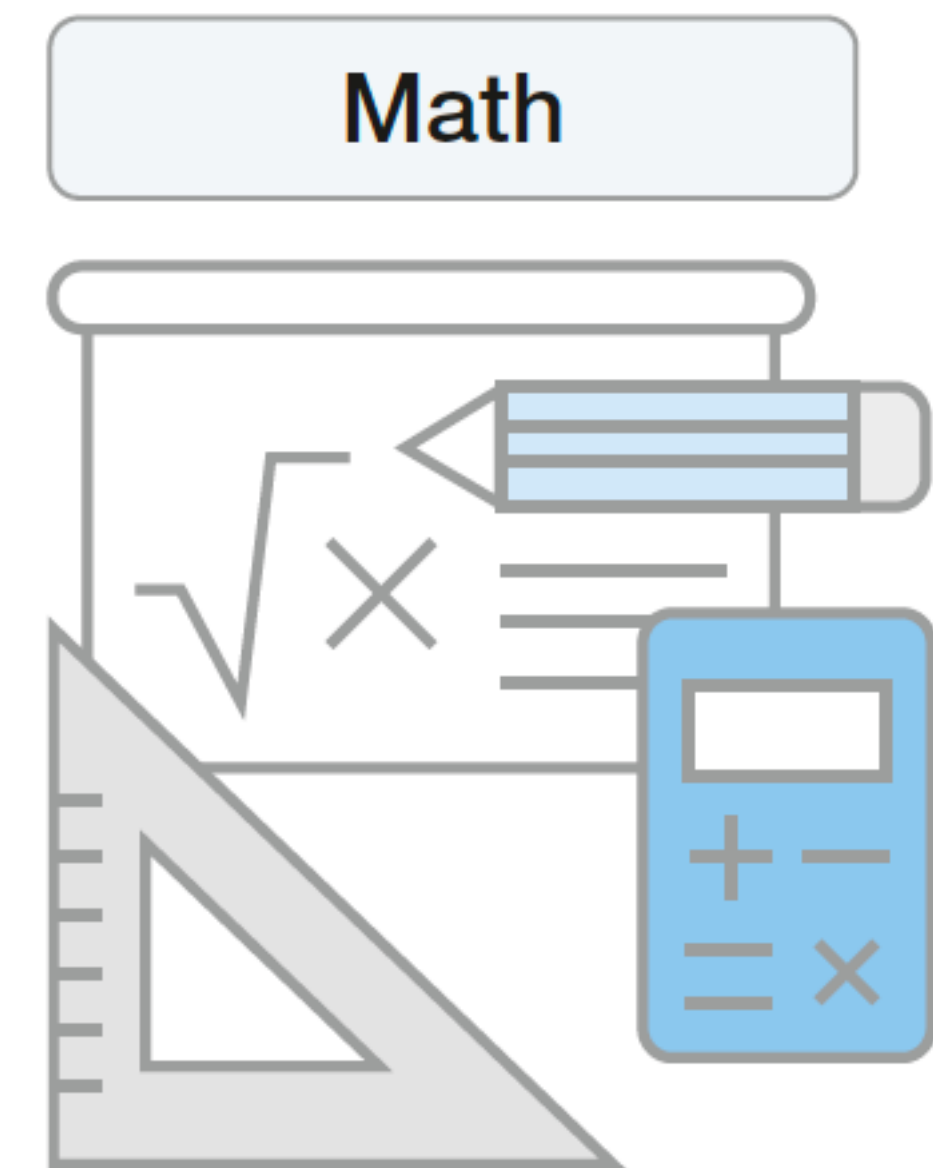


그림 9-6 Math 클래스

실습 9-2

Math 클래스의 기본 활용

ch09/ex9_2/MathTest.java

```
01 import java.lang.Math;
02
03 public class MathTest {
04     public static void main(String[] args) {
05         System.out.printf("수학의 파이(원주율) 값: %f\n", Math.PI);
06         System.out.printf("임의의 난수 값: %f\n", Math.random());
07         System.out.printf("9.81의 내림값: %f\n", Math.floor(9.81));
08         System.out.printf("4의 제곱근: %f\n", Math.sqrt(4));
09         System.out.printf("2의 3승: %f\n", Math.pow(2, 3));
10     }
11 }
```

API 불러오기(생략 가능)

② 임의의 실숫값 반환

① 원주율 변수

③ 실숫값
내림 처리

④ 제곱근 구하기

⑤ 2의 3 제곱 계산하기

Math 클래스를 활용한 프로그램과 그 출력 결과

수학의 파이(원주율) 값: 3.141593
임의의 난수 값: 0.016996
9.81의 내림값: 9.000000
4의 제곱근: 2.000000
2의 3승: 8.000000

실습 코드

https://github.com/cloudstudyin-g-kr/JavaPlz/blob/master/src/part2/ch09/ex9_2/MathTest.java

실습 9-2

Math 클래스의 기본 활용

ch09/ex9_2/MathTest.java

```
01 import java.lang.Math;
02
03 public class MathTest {
04     public static void main($
05         System.out.printf("수학의 파이(원주율) 값: %f\n", Math.PI);
06         System.out.printf("임의 난수 값: %f\n", Math.random());
07         System.out.printf("9.81의 내림값: %f\n", Math.floor(9.81));
08         System.out.printf("4의 제곱근: %f\n", Math.sqrt(4));
09         System.out.printf("2의 3승: %f\n", Math.pow(2, 3));
10     }
11 }
```

API 불러오기
java.lang 패키지의 Math 클래스를 불러옴
- java.lang 패키지는 가장 기본이 되는 패키지
로 자동으로 import 되기 때문에 생략 가능

③ 실숫값
내림 처리

④ 제곱근 구하기

⑤ 2의 3 제곱 계산하기

실행 결과

수학의 파이(원주율) 값: 3.141593
임의 난수 값: 0.016996
9.81의 내림값: 9.000000
4의 제곱근: 2.000000
2의 3승: 8.000000

실습 9-2

Math 클래스의 기본 활용

ch09/ex9_2/MathTest.java

```
01 import java.lang.Math;
02
03 public class MathTest {
04     public static void main(String[] args) {
05         System.out.printf("수학의 파이(원주율) 값: %f\n", Math.PI);
06         System.out.printf("임의 난수 값: %f\n", Math.random());
07         System.out.printf("9.81의 내림값: %f\n", Math.floor(9.81));
08         System.out.printf("4의 제곱근: %f\n", Math.sqrt(4));
09         System.out.printf("2의 3승: %f\n", Math.pow(2, 3));
10     }
11 }
```

API 불러오기(생략 가능)

원주율 변수
수학에서 사용되는 원주율(r: 파이) 값을 지닌 변수

③ 실숫값
내림 처리

④ 제곱근 구하기

⑤ 2의 3 제곱 계산하기

실행 결과

수학의 파이(원주율) 값: 3.141593
임의 난수 값: 0.016996
9.81의 내림값: 9.000000
4의 제곱근: 2.000000
2의 3승: 8.000000

실습 9-2

Math 클래스의 기본 활용

ch09/ex9_2/MathTest.java

```
01 import java.lang.Math;
02
03 public class MathTest {
04     public static void main(String[] args) {
05         System.out.printf("수학의 파이(원주율) 값: %f\n", Math.PI);
06         System.out.printf("임의의 난수 값: %f\n", Math.random());
07         System.out.printf("9.81의 내림값: %f\n", Math.floor(9.81));
08         System.out.printf("4의 제곱근: %f\n", Math.sqrt(4));
09         System.out.printf("2의 3승: %f\n", Math.pow(2, 3));
10     }
11 }
```

API 불러오기(생략 가능)

② 임의의 실숫값 반환

① 원주율 변수

임의의 실숫값 생성 메소드
0.0 이상 1.0 미만의 실숫값을 반환하는 메소드

④ 제곱근 구하기

⑤ 2의 3 제곱 계산하기

실행 결과

수학의 파이(원주율) 값: 3.141593
임의의 난수 값: 0.016996
9.81의 내림값: 9.000000
4의 제곱근: 2.000000
2의 3승: 8.000000

실습 9-2

Math 클래스의 기본 활용

ch09/ex9_2/MathTest.java

```
01 import java.lang.Math;
02
03 public class MathTest {
04     public static void main(String[] args) {
05         System.out.printf("수학의 파이(원주율) 값: %f\n", Math.PI);
06         System.out.printf("임의 난수 값: %f\n", Math.random());
07         System.out.printf("9.81의 내림값: %f\n", Math.floor(9.81));
08         System.out.printf("4의 제곱근: %f\n", Math.sqrt(4));
09         System.out.printf("2의 3승: %f\n", Math.pow(2, 3));
10     }
11 }
```

API 불러오기(생략 가능)

② 임의의 실숫값 반환

① 원주율 변수

③ 실숫값

내림값 계산 메소드
입력값 9.81을 내림처리하여 9.00을 반환

⑤ 2의 3 제곱 계산하기

실행 결과

수학의 파이(원주율) 값: 3.141593
임의 난수 값: 0.016996
9.81의 내림값: 9.000000
4의 제곱근: 2.000000
2의 3승: 8.000000

실습 9-2

Math 클래스의 기본 활용

ch09/ex9_2/MathTest.java

```
01 import java.lang.Math;
02
03 public class MathTest {
04     public static void main(String[] args) {
05         System.out.printf("수학의 파이(원주율) 값: %f\n", Math.PI);
06         System.out.printf("임의의 난수 값: %f\n", Math.random());
07         System.out.printf("9.81의 내림값: %f\n", Math.floor(9.81));
08         System.out.printf("4의 제곱근: %f\n", Math.sqrt(4));
09         System.out.printf("2의 3승: %f\n", Math.pow(2, 3));
10     }
11 }
```

API 불러오기(생략 가능)

② 임의의 실숫값 반환

① 원주율 변수

③ 실숫값
내림 처리

제곱근 계산 메소드
입력값 4의 제곱근을 실수 2.0으로 반환

실행 결과

수학의 파이(원주율) 값: 3.141593
임의의 난수 값: 0.016996
9.81의 내림값: 9.000000
4의 제곱근: 2.000000
2의 3승: 8.000000

실습 9-2

Math 클래스의 기본 활용

ch09/ex9_2/MathTest.java

```
01 import java.lang.Math;
02
03 public class MathTest {
04     public static void main(String[] args) {
05         System.out.printf("수학의 파이(원주율) 값: %f\n", Math.PI);
06         System.out.printf("임의 난수 값: %f\n", Math.random());
07         System.out.printf("9.81의 내림값: %f\n", Math.floor(9.81));
08         System.out.printf("4의 제곱근: %f\n", Math.sqrt(4));
09         System.out.printf("2의 3승: %f\n", Math.pow(2, 3));
10     }
11 }
```

API 불러오기(생략 가능)

② 임의의 실숫값 반환

① 원주율 변수

③ 실숫값
내림 처리

④ 제곱근 구하기

거듭 제곱 계산 메소드

입력값 2의 3승(거듭제곱)을 실수 8.0으로 반환

실행 결과

수학의 파이(원주율) 값: 3.141593
임의 난수 값: 0.016996
9.81의 내림값: 9.000000
4의 제곱근: 2.000000
2의 3승: 8.000000

실습 9-2

Math 클래스의 기본 활용

ch09/ex9_2/MathTest.java

```
01 import java.lang.Math;
02
```

-----API 불러오기(생략 가능)

표 9-1 Math 클래스 설명

코드	설명
static double PI	수학에서 사용되는 원주율(π : 파이) 값을 지닌 변수
static double random()	0.0 이상 1.0 미만의 실숫값을 반환하는 메소드
static double floor(double a)	실수 a의 내림값을 반환하는 메소드
static double sqrt(double a)	실수 a의 제곱근을 반환하는 메소드
static double pow(double a, double b)	실수 a를 b로 거듭제곱한 결과를 반환하는 메소드

수학의 파이(원주율) 값: 3.141593
임의 난수 값: 0.016996
9.81의 내림값: 9.000000
4의 제곱근: 2.000000
2의 3승: 8.000000

실습 9-2

Math 클래스의 기본 활용

ch09/ex9_2/MathTest.java

```
01 import java.lang.Math;-----API 불러오기(생략 가능)
```

Math 클래스의 자세한 내용은 Java API 문서를 참조

random

public static double random()

Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0. Returned values are chosen pseudorandomly with (approximately) uniform distribution from that range.

When this method is first called, it creates a single new pseudorandom-number generator, exactly as if by the expression

new java.util.Random()

This new pseudorandom-number generator is used thereafter for all calls to this method and is used nowhere else.

This method is properly synchronized to allow correct use by more than one thread. However, if many threads need to generate pseudorandom numbers at a great rate, it may reduce contention for each thread to have its own pseudorandom-number generator.

API Note:

As the largest double value less than 1.0 is Math.nextDown(1.0), a value x in the closed range [x1,x2] where x1<=x2 may be defined by the statements

double f = Math.random()/Math.nextDown(1.0);
double x = x1*(1.0 - f) + x2*f;

Returns:

a pseudorandom double greater than or equal to 0.0 and less than 1.0.

See Also:

nextDown(double), Random.nextDouble()

그림 9-7 자바 API 문서의 Math.random() 부분 [https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/Math.html#random\(\)](https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/lang/Math.html#random())

2의 3승: 8.000000

II . Random 클래스

Random 클래스는 난수 생성을 위한 자바 API로,

- 임의의 값을 쉽게 만들어줌
- java.util 패키지에 위치
- Math.random()과 달리 객체를 만들어 사용해야 함

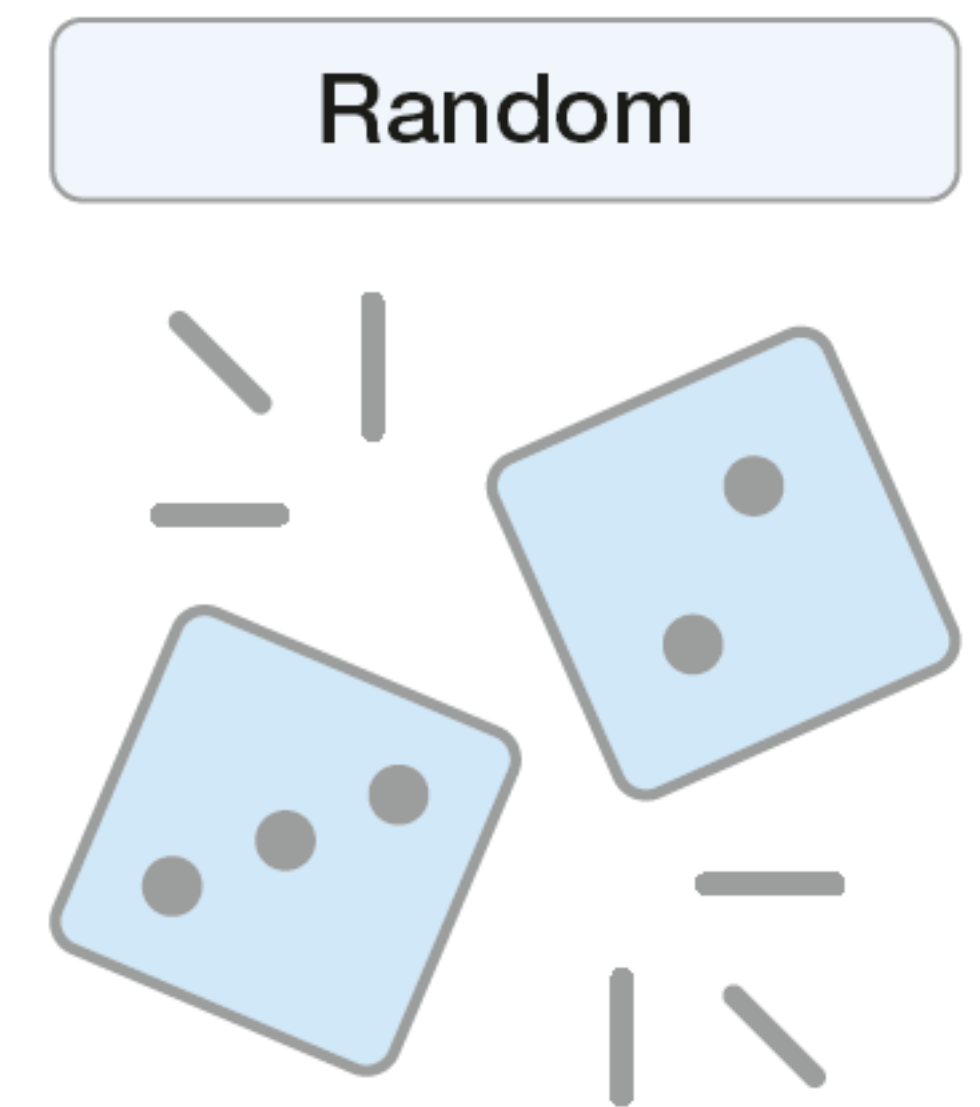


그림 9-8 Random 클래스

실습 9-3

Random 클래스 기본 활용

ch09/ex9_3/RandomTest.java

```
01 import java.util.Random;-----API 불러오기
02
03 public class RandomTest {
04     public static void main(String[] args) {
05         Random random = new Random();-----Random 객체 생성
06         int n = 10;
07
08         System.out.printf("임의의 정수: %d\n", random.nextInt());
09         System.out.printf("0이상 N(%d)미만의 임의 정수: %d\n", n,
10                             random.nextInt(n));-----② 0 이상 n 미만의 임의 정수 생성
11         System.out.printf("임의의 실수: %f\n", random.nextDouble());
12         System.out.printf("임의의 참/거짓: %b\n", random.nextBoolean());
13     }-----③ 임의의 실숫값 반환-----④ 임의의 참/거짓 반환
```

실행 결과

임의의 정수: -2040144020
0이상 N(10)미만의 임의 정수: 4
임의의 실수: 0.329775
임의의 참/거짓: true

실습 코드

https://github.com/cloudstudyin-g-kr/JavaPlz/blob/master/src/part2/ch09/ex9_3/RandomTest.java

실습 9-3

Random 클래스 기본 활용

ch09/ex9_3/RandomTest.java

```
01 import java.util.Random;
02
03 public class RandomTest {
04     public static void main(String[] args) {
05         Random random = new Random();
06         int n = 10;
07
08         System.out.printf("임의의 정수: %d\n", random.nextInt());
09         System.out.printf("0이상 N(%d)미만의 임의 정수: %d\n", n,
10                             random.nextInt(n));
11         System.out.printf("임의의 실수: %f\n", random.nextDouble());
12         System.out.printf("임의의 참/거짓: %b\n", random.nextBoolean());
13     }
14 }
```

API 불러오기
java.util 패키지의 Random 클래스를 불러옴

Random 객체 생성

① 임의의 정숫값 반환

② 0 이상 n 미만의 임의 정수 생성

③ 임의의 실숫값 반환

④ 임의의 참/거짓 반환

실행 결과

임의의 정수: -2040144020
0이상 N(10)미만의 임의 정수: 4
임의의 실수: 0.329775
임의의 참/거짓: true

실습 9-3

Random 클래스 기본 활용

ch09/ex9_3/RandomTest.java

```
01 import java.util.Random;-----API 불러오기
02
03 public class RandomTest {
04     public static void main(String[] args) {
05         Random random = new Random();-----Random 객체 생성
06         int n = 10;-----생성자 호출로 Random 객체를 만듦
07
08         System.out.printf("임의의 정수: %d\n", random.nextInt());-----① 임의의 정숫값 반환
09         System.out.printf("0이상 N(%d)미만의 임의 정수: %d\n", n,
10                             random.nextInt(n));-----② 0 이상 n 미만의 임의 정수 생성
11         System.out.printf("임의의 실수: %f\n", random.nextDouble());
12         System.out.printf("임의의 참/거짓: %b\n", random.nextBoolean());
13     }-----③ 임의의 실숫값 반환-----④ 임의의 참/거짓 반환
```

실행 결과

임의의 정수: -2040144020
0이상 N(10)미만의 임의 정수: 4
임의의 실수: 0.329775
임의의 참/거짓: true

실습 9-3

Random 클래스 기본 활용

ch09/ex9_3/RandomTest.java

```
01 import java.util.Random;
02
03 public class RandomTest {
04     public static void main(String[] args) {
05         Random random = new Random();
06         int n = 10;
07
08         System.out.printf("임의의 정수: %d\n", random.nextInt());
09         System.out.printf("0이상 N(%d)미만의 임의 정수: %d\n", n,
10                             random.nextInt(n));
11         System.out.printf("임의의 실수: %f\n", random.nextDouble());
12         System.out.printf("임의의 참/거짓: %b\n", random.nextBoolean());
13     }
14 }
```

API 불러오기

Random 객체 생성

임의의 정숫값 생성 메소드
int의 표현 범위 내 임의의 정숫값을 반환

② 0 이상 n 미만의 임의 정수 생성

③ 임의의 실숫값 반환

④ 임의의 참/거짓 반환

실행 결과

임의의 정수: -2040144020
0이상 N(10)미만의 임의 정수: 4
임의의 실수: 0.329775
임의의 참/거짓: true

실습 9-3

Random 클래스 기본 활용

ch09/ex9_3/RandomTest.java

```
01 import java.util.Random;-----API 불러오기
02
03 public class RandomTest {
04     public static void main(String[] args) {
05         Random random = new Random();-----Random 객체 생성
06         int n = 10;
07
08         System.out.printf("임의의 정수: %d\n", random.nextInt());
09         System.out.printf("0이상 N(%d)미만의 임의 정수: %d\n", n,
10                             random.nextInt(n));
11         System.out.printf("임의의 실수: %f\n", random.nextDouble());
12         System.out.printf("임의의 참/거짓: %b\n", random.nextBoolean());
13     }
14 }
```

① 임의의 정숫값 반환

임의의 정숫값 생성 메소드(2)
0 이상 n 미만의 임의의 정숫값을 반환

③ 임의의 실숫값 반환

④ 임의의 참/거짓 반환

실행 결과

임의의 정수: -2040144020
0이상 N(10)미만의 임의 정수: 4
임의의 실수: 0.329775
임의의 참/거짓: true

실습 9-3

Random 클래스 기본 활용

ch09/ex9_3/RandomTest.java

```

01 import java.util.Random; ----- API 불러오기
02
03 public class RandomTest {
04     public static void main(String[] args) {
05         Random random = new Random(); ----- Random 객체 생성
06         int n = 10;
07
08         System.out.printf("임의의 정수: %d\n", random.nextInt());
09         System.out.printf("0이상 N(%d)미만의 임의 정수: %d\n", n,
10                             random.nextInt(n)); ----- ① 임의의 정숫값 반환
11         System.out.printf("임의의 실수: %f\n", random.nextDouble()); ----- ② 0 이상 n 미만의 임의 정수 생성
12         System.out.printf("임의의 참/거짓: %b\n", random.nextBoolean());
13     } ----- ③ 임의의 실숫값 반환

```

임의의 실숫값 생성 메소드

0.0 이상 1.0 미만의 임의의 실숫값을 반환

실행 결과

```

임의의 정수: -2040144020
0이상 N(10)미만의 임의 정수: 4
임의의 실수: 0.329775
임의의 참/거짓: true

```

실습 9-3

Random 클래스 기본 활용

ch09/ex9_3/RandomTest.java

```
01 import java.util.Random;-----API 불러오기
02
03 public class RandomTest {
04     public static void main(String[] args) {
05         Random random = new Random();-----Random 객체 생성
06         int n = 10;
07
08         System.out.printf("임의의 정수: %d\n", random.nextInt());
09         System.out.printf("0이상 N(%d)미만의 임의 정수: %d\n", n,
10                             random.nextInt(n));-----① 임의의 정숫값 반환
11         System.out.printf("임의의 실수: %f\n", random.nextDouble());
12         System.out.printf("임의의 참/거짓: %b\n", random.nextBoolean());
13     }-----② 0 이상 n 미만의 임의 정수 생성
14 }
```

임의의 논릿값 생성 메소드
참(true) 또는 거짓(false)을 임의 반환

실행 결과

임의의 정수: -2040144020
0이상 N(10)미만의 임의 정수: 4
임의의 실수: 0.329775
임의의 참/거짓: true

실습 9-3

Random 클래스 기본 활용

ch09/ex9_3/RandomTest.java

01

import java.util.Random;

-----API 불러오기

02

표 9-2 Random 클래스 설명 <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/Random.html>

코드	설명
Random()	Random 클래스의 생성자
int nextInt()	int의 표현 범위 내 임의의 정숫값을 반환하는 메소드
int nextInt(int n)	0 이상 n 미만의 임의의 정숫값을 반환하는 메소드
double nextDouble()	0.0 이상 1.0 미만인 임의의 실숫값을 반환하는 메소드
boolean nextBoolean()	참 _{true} 또는 거짓 _{false} 을 임의 반환하는 메소드

실행 결과

임의의 정수: -2040144020
0이상 N(10)미만의 임의 정수: 4
임의의 실수: 0.329775
임의의 참/거짓: true

III. ArrayList 클래스

ArrayList 클래스는 객체들을 저장할 수 있는 자바 API로,

- 객체 관리를 돕는 배열의 또 다른 형태
- java.util 패키지에 위치

! TIP ArrayList와 같이 여러 데이터의 효율적 관리를 위한 하나의 틀을 **자료구조**(data structure)라 함.

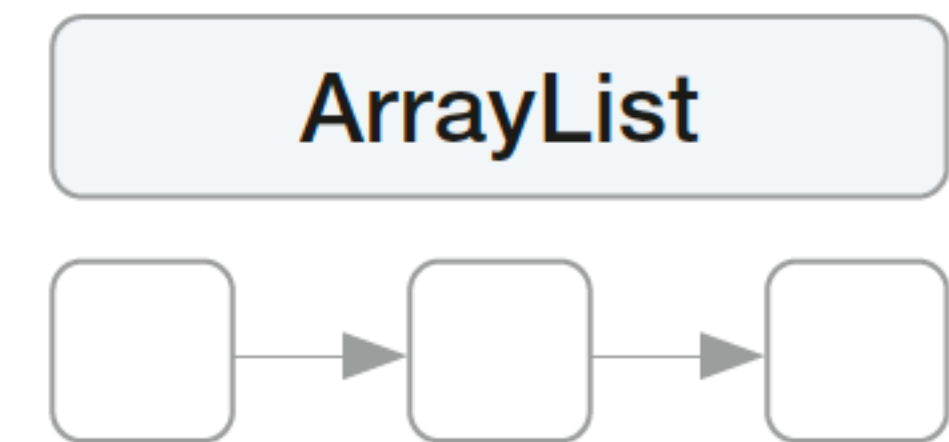


그림 9-9 ArrayList 클래스

실습 9-4ArrayList 클래스 기본 활용ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;-----API 불러오기
02
03 public class ArrayListTest {
04     public static void main(String[] args) {-----String 저장을 위한
05         ArrayList<String> names = new ArrayList<String>();-----ArrayList 객체 생성
06
07         names.add("Kim");-----String 데이터 추가
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");-----0번 인덱스 데이터를 "Han"으로 변경
13         String removed = names.remove(1);-----1번 인덱스 데이터 제거 후, 해당 값 반환
14
15         for (int i = 0; i < names.size(); i++) {-----ArrayList에 담긴 데이터의 개수 반환
16             System.out.printf("%s ", names.get(i));-----i번 인덱스 값을 반환
17         }
18     }
19 }
```

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기
java.util 패키지의 ArrayList 클래스를 불러옴

String 저장할 위한
ArrayList 객체 생성

String 데이터 추가

0번 인덱스 데이터를 "Han"으로 변경

1번 인덱스 데이터 제거 후, 해당 값 반환

ArrayList에 담긴 데이터의 개수 반환

i번 인덱스 값을 반환

실습 9-4ArrayList 클래스 기본 활용ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;-----API 불러오기
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");-----String 데이터 추가
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         -----0번 인덱스 데이터를 "Han"으로 변경
13         names.set(0, "Han");-----1번 인덱스 데이터 제거 후, 해당 값 반환
14         String removed = names.remove(1);
15         -----ArrayList에 담긴 데이터의 개수 반환
16         for (int i = 0; i < names.size(); i++) {
17             System.out.printf("%s ", names.get(i));
18         }-----i번 인덱스 값을 반환
19     }
```

ArrayList 객체 생성
문자열(String) 저장을 위한 ArrayList 객체 생성

실행 결과

Han Park Choi

실습 9-4

ArrayList 클래스 기본 활용

ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

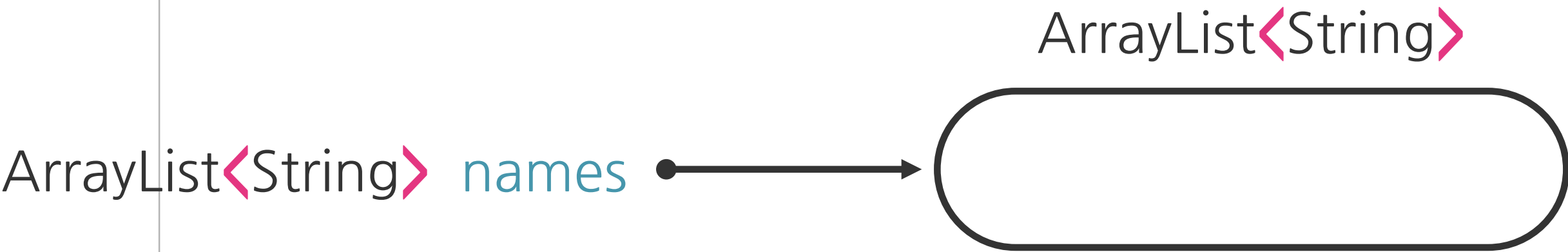
ArrayList 객체 생성
문자열(String) 저장을 위한 ArrayList 객체 생성

TIP ArrayList의 경우 꺾쇠 안쪽의 값에 따라 관리할 객체의 타입이 정해짐. 이러한 기법을 제네릭 (generic)이라고 함.

실행 결과

Han Park Choi


```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```



실행 결과

Han Park Choi

실습 9-4

ArrayList 클래스 기본 활용

ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기

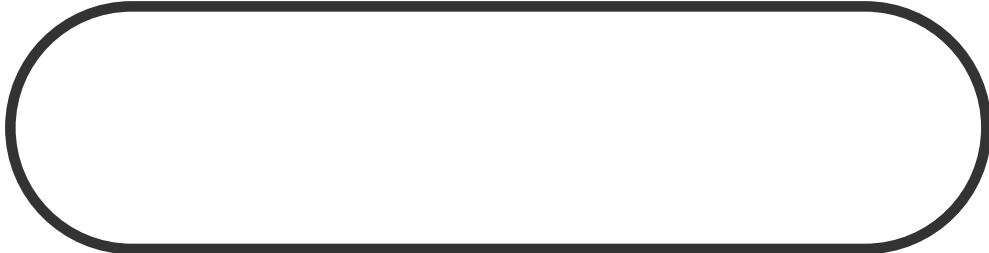
String 저장을 위한
ArrayList 객체 생성

ArrayList<String> names

String 데이터 추가
문자열 "Kim"을 ArrayList 객체의 마지막 위치에 추가

0번 인덱스 데이터를 "Han"으로 변경
1번 인덱스 데이터 제거 후, 해당 값 반환
ArrayList에 담긴 데이터의 개수 반환
i번 인덱스 값을 반환

ArrayList<String>



실행 결과

Han Park Choi

실습 9-4

ArrayList 클래스 기본 활용

ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

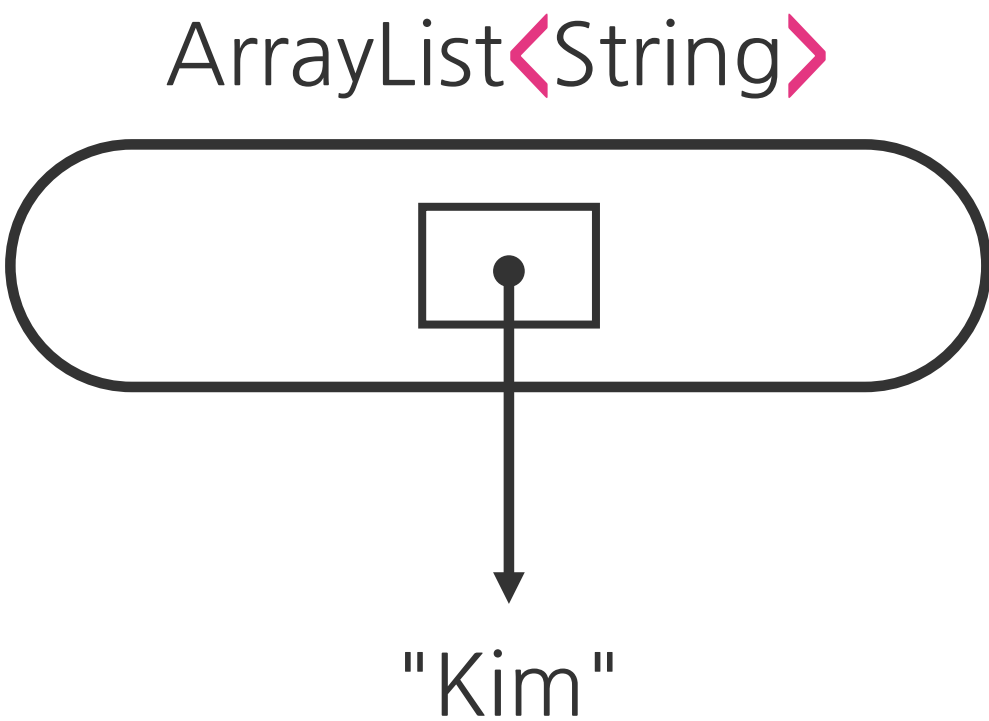
API 불러오기

String 저장을 위한
ArrayList 객체 생성

ArrayList<String> names

String 데이터 추가
문자열 "Kim"을 ArrayList 객체의 마지막 위치에 추가

0번 인덱스 데이터를 "Han"으로 변경
1번 인덱스 데이터 제거 후, 해당 값 반환
ArrayList에 담긴 데이터의 개수 반환
i번 인덱스 값을 반환



실행 결과

Han Park Choi

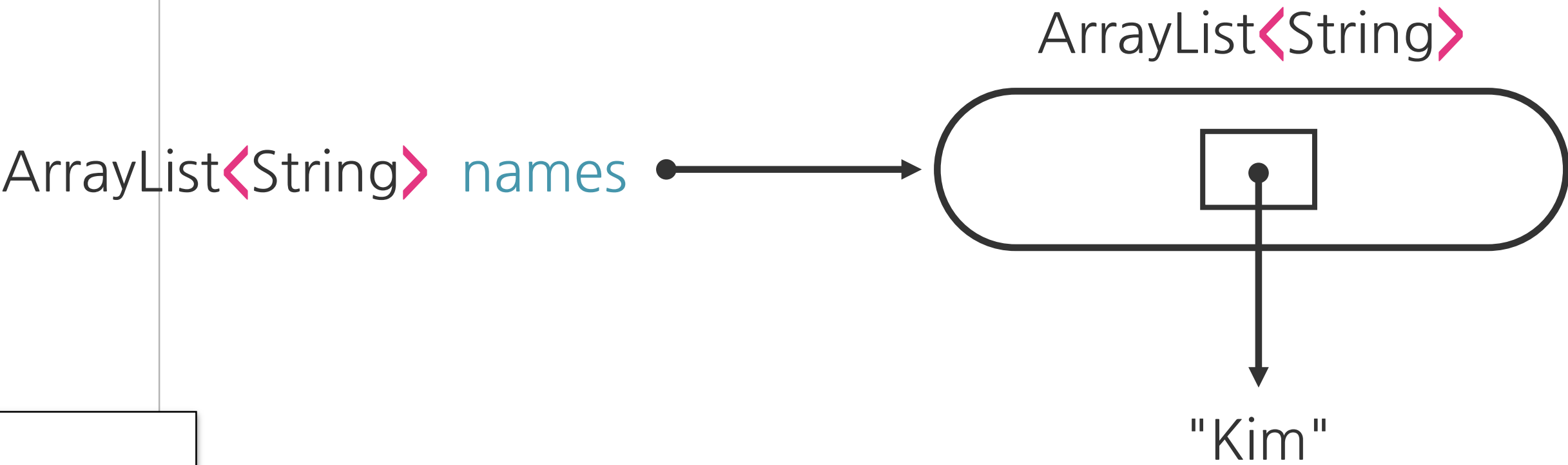
실습 9-4

ArrayList 클래스 기본 활용

ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09
10         names.set(0, "Han");
11         String removed = names.remove(1);
12
13         for (int i = 0; i < names.size(); i++) {
14             System.out.printf("%s ", names.get(i));
15         }
16     }
17 }
18
19 }
```

String 데이터 추가
문자열 "Lee"을 ArrayList 객체의 마지막 위치에 추가



실행 결과

Han Park Choi

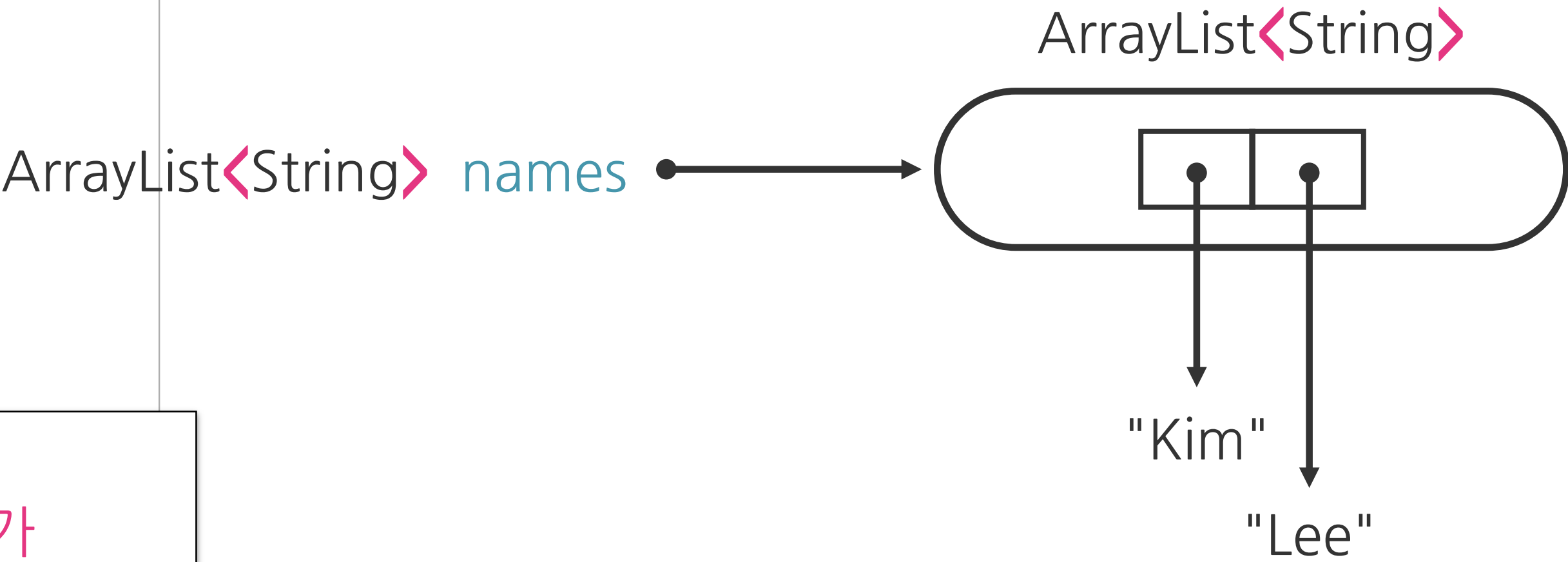
실습 9-4

ArrayList 클래스 기본 활용

ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09
10         names.set(0, "Han");
11         String removed = names.remove(1);
12
13         for (int i = 0; i < names.size(); i++) {
14             System.out.printf("%s ", names.get(i));
15         }
16     }
17 }
18
19 }
```

String 데이터 추가
문자열 "Lee"을 ArrayList 객체의 마지막 위치에 추가



실행 결과

Han Park Choi

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

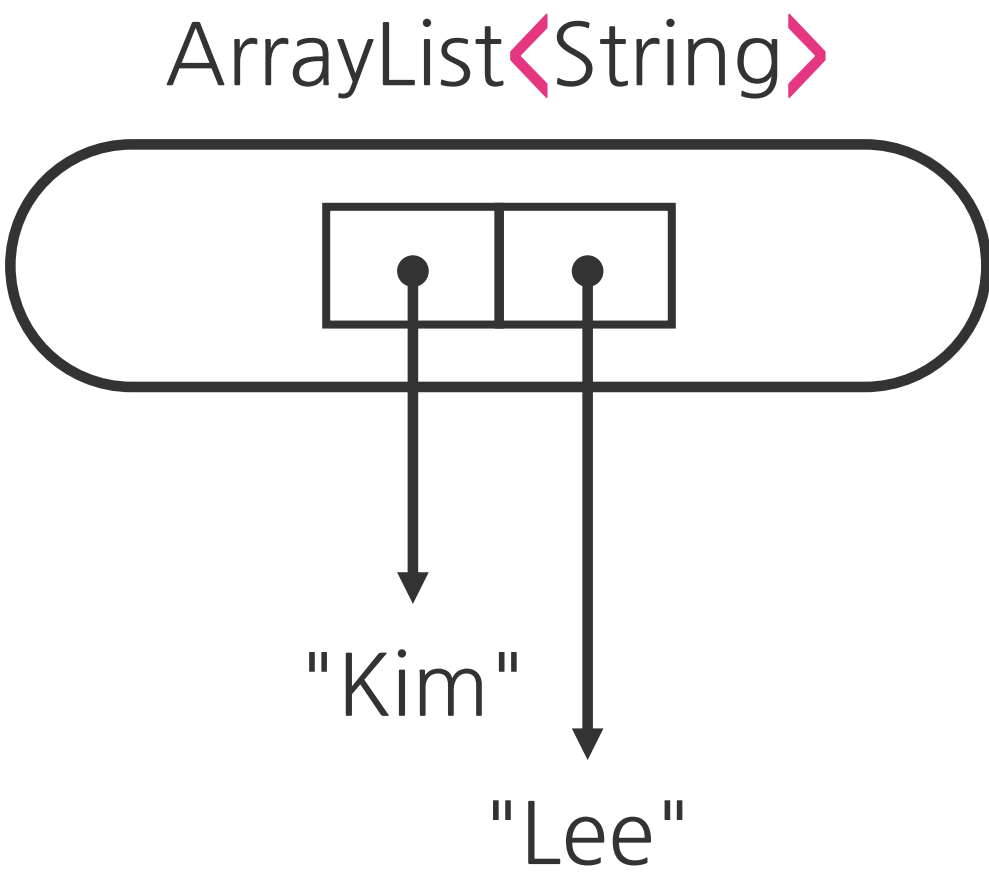
API 불러오기

String 저장을 위한
ArrayList 객체 생성

String 데이터 추가

String 데이터 추가
"Park"과 "Choi"를 ArrayList 객체의 마지막 위치에 차례로 추가

ArrayList<String> names



실행 결과

Han Park Choi


```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

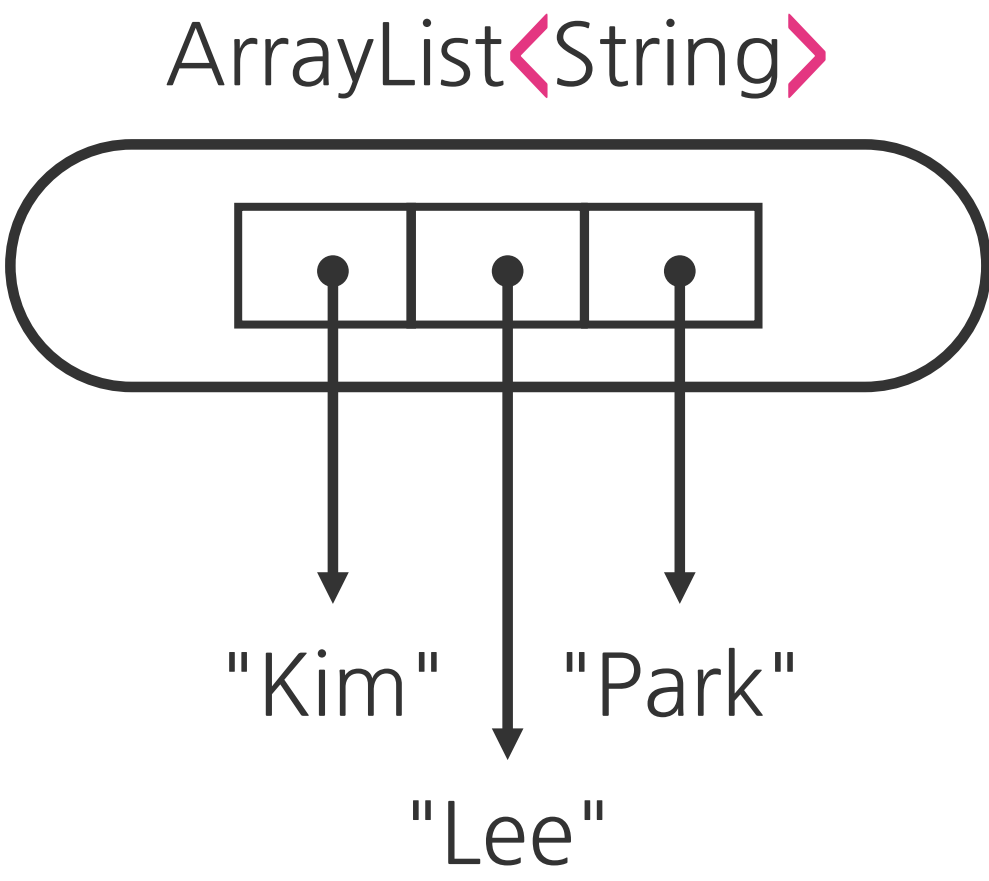
API 불러오기

String 저장을 위한
ArrayList 객체 생성

String 데이터 추가

String 데이터 추가
"Park"과 "Choi"를 ArrayList 객체의 마지막 위치에 차례로 추가

ArrayList<String> names



실행 결과

Han Park Choi


```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

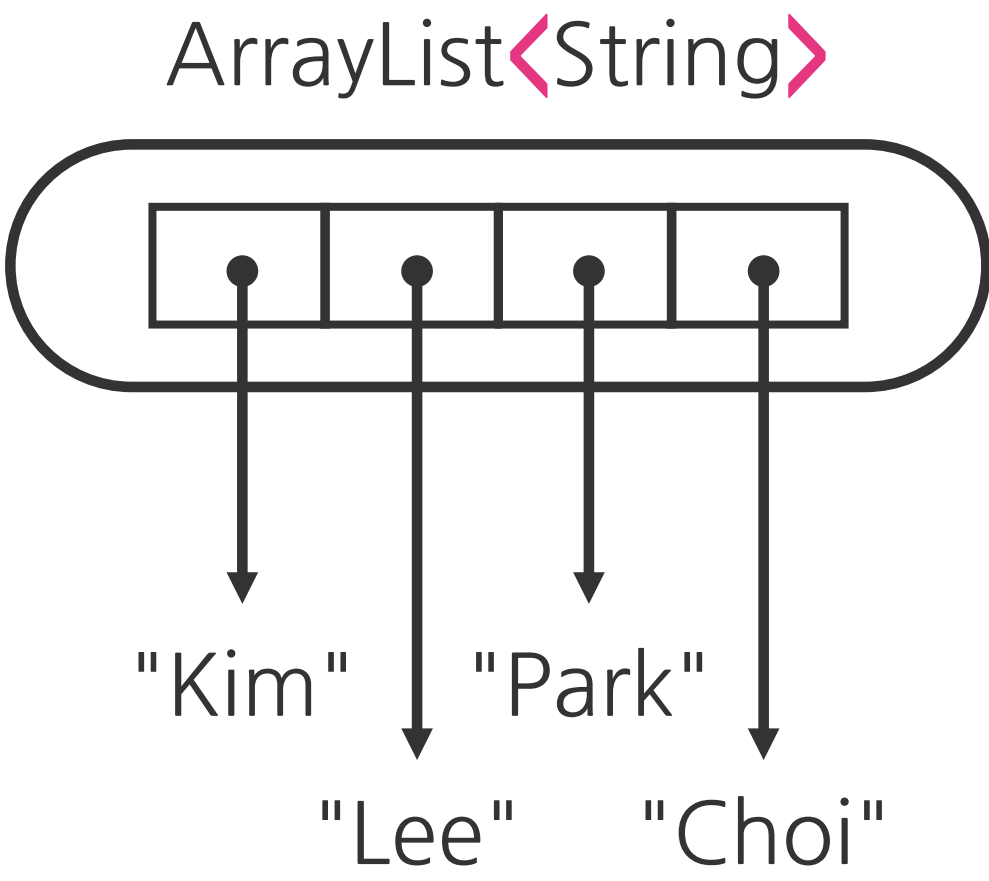
API 불러오기

String 저장을 위한
ArrayList 객체 생성

String 데이터 추가

String 데이터 추가
"Park"과 "Choi"를 ArrayList 객체의 마지막 위치에 차례로 추가

ArrayList<String> names

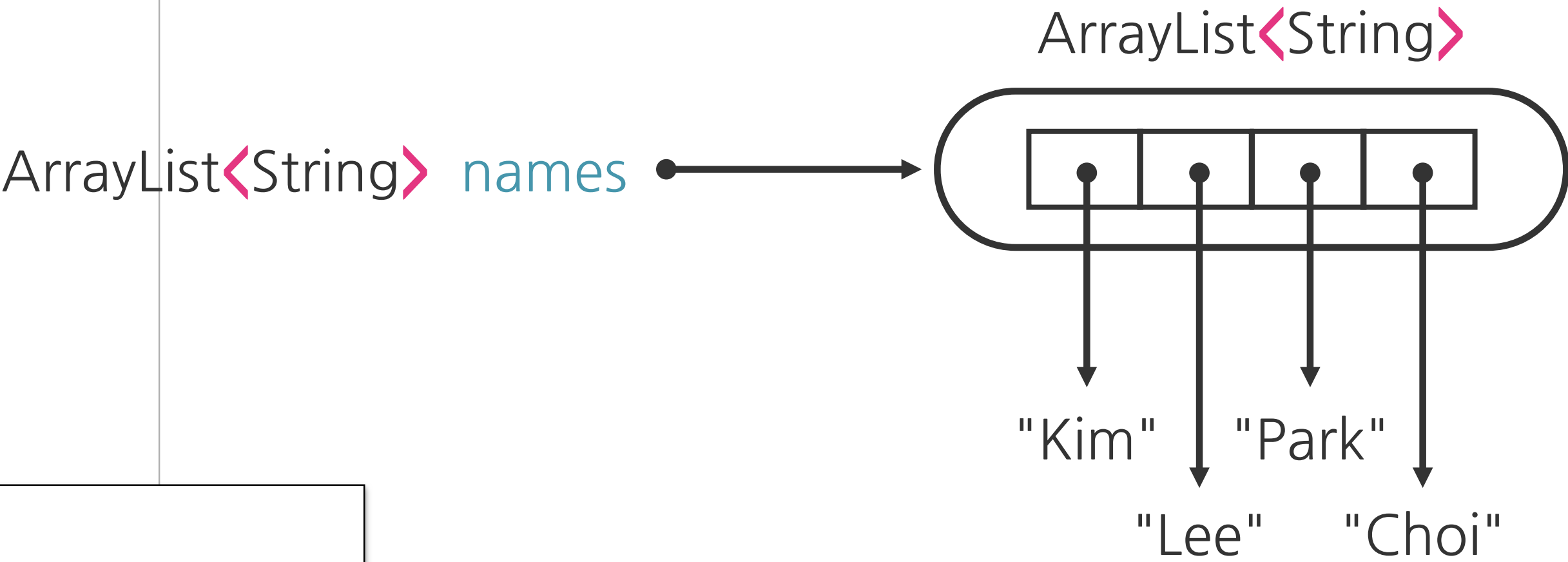


실행 결과

Han Park Choi

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

특정 인덱스 요소 수정
0번 인덱스의 문자열 "Kim"이 "Han"으로 변경



실행 결과

Han Park Choi

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

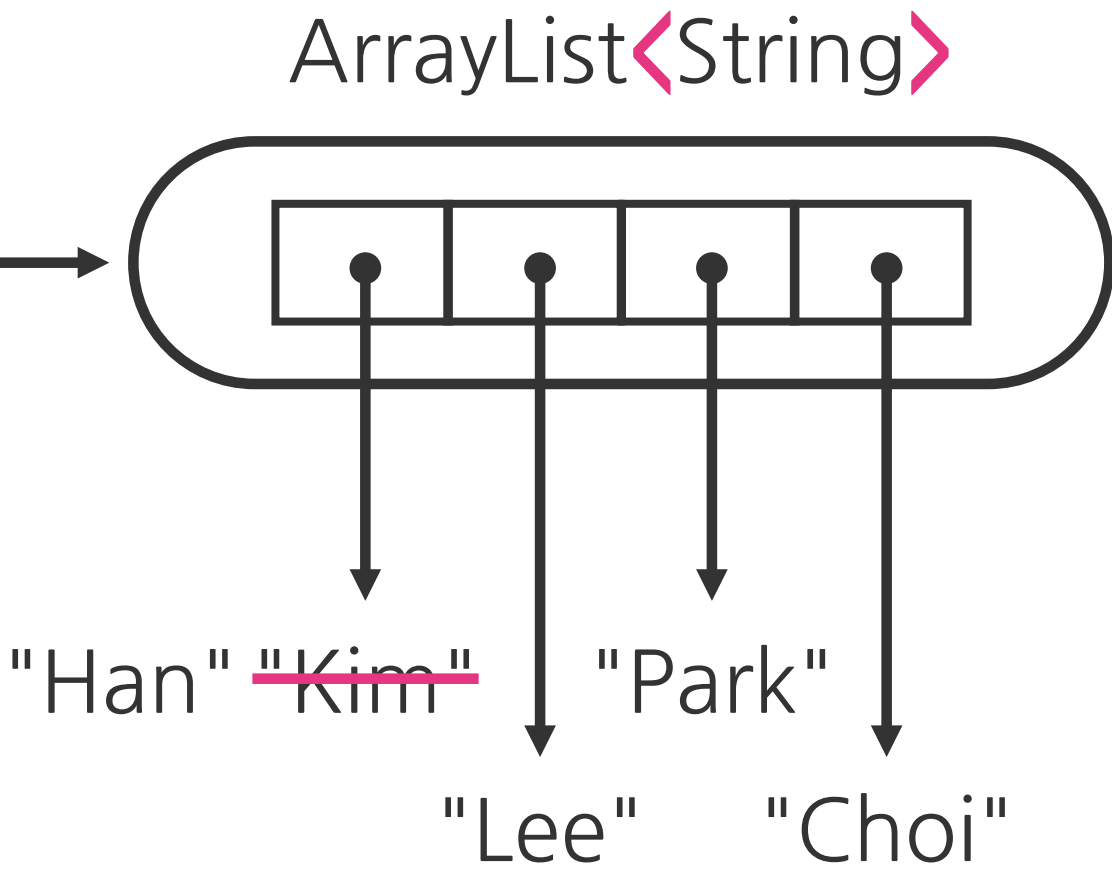
API 불러오기

String 저장을 위한
ArrayList 객체 생성

String 데이터 추가

특정 인덱스 요소 수정
0번 인덱스의 문자열 "Kim"이 "Han"으로 변경

ArrayList<String> names



실행 결과

Han Park Choi

실습 9-4

ArrayList 클래스 기본 활용

ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기

String 저장을 위한
ArrayList 객체 생성

String 데이터 추가

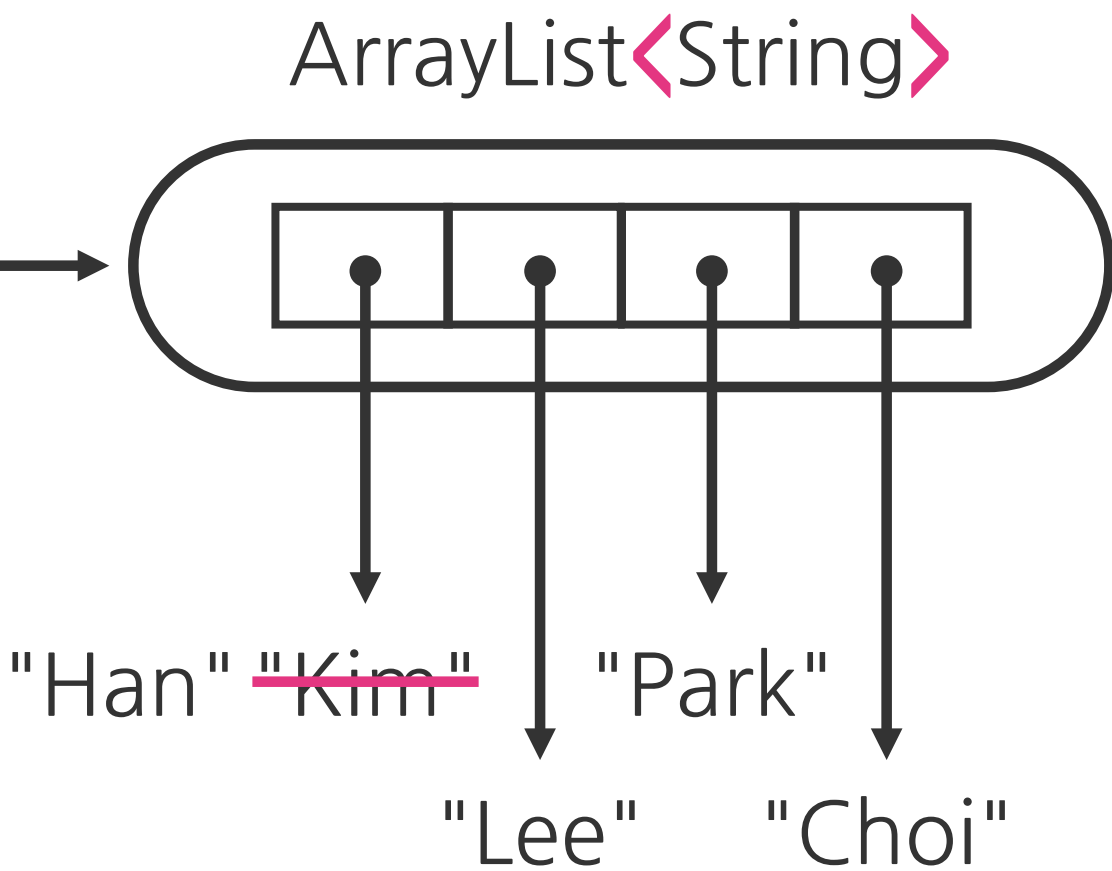
0번 인덱스 데이터를 "Han"으로 변경

1번

ArrayList

i번 인덱스 값을 반환

ArrayList<String> names



특정 인덱스 요소 삭제 및 반환
1번 인덱스의 "Lee"가 ArrayList에서 제거됨과 동시에 반환

실행 결과

Han Park Choi


```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기

String 저장을 위한
ArrayList 객체 생성

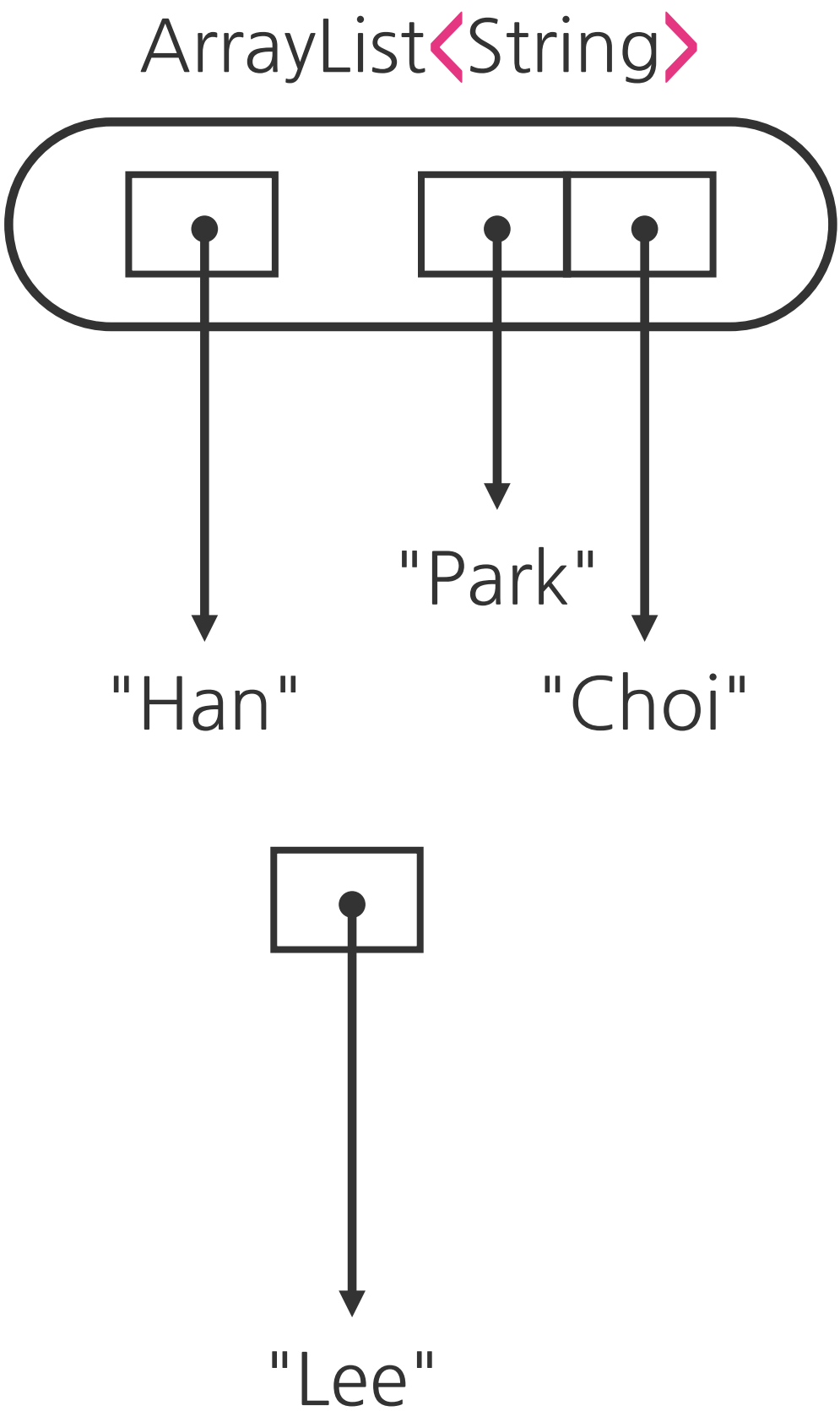
String 데이터 추가

0번 인덱스 데이터를 "Han"으로 변경

특정 인덱스 요소 삭제 및 반환
1번 인덱스의 "Lee"가 ArrayList에서 제거됨과 동시에 반환

i번 인덱스 값을 반환

ArrayList<String> names



실행 결과

Han Park Choi

실습 9-4 ArrayList 클래스 기본 활용 ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기

String 저장을 위한
ArrayList 객체 생성

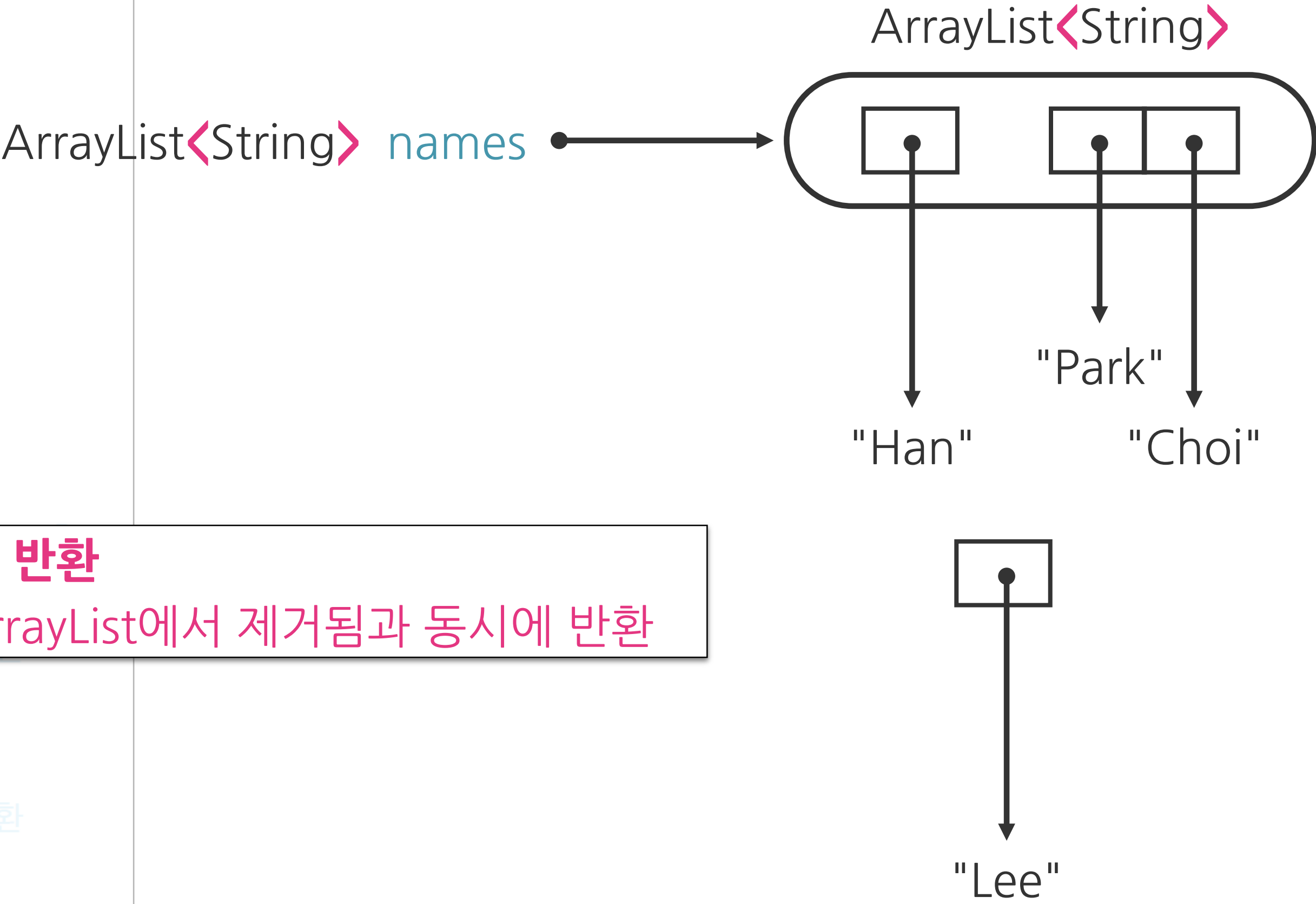
String 데이터 추가

0번 인덱스 데이터를 "Han"으로 변경

특정 인덱스 요소 삭제 및 반환
1번 인덱스의 "Lee"가 ArrayList에서 제거됨과 동시에 반환

i번 인덱스 값을 반환

remove() 메소드 수행 시
삭제 인덱스의 데이터는 한 칸씩 당겨져 빈자리를 채움



실행 결과

Han Park Choi

실습 9-4

ArrayList 클래스 기본 활용

ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기

String 저장을 위한
ArrayList 객체 생성

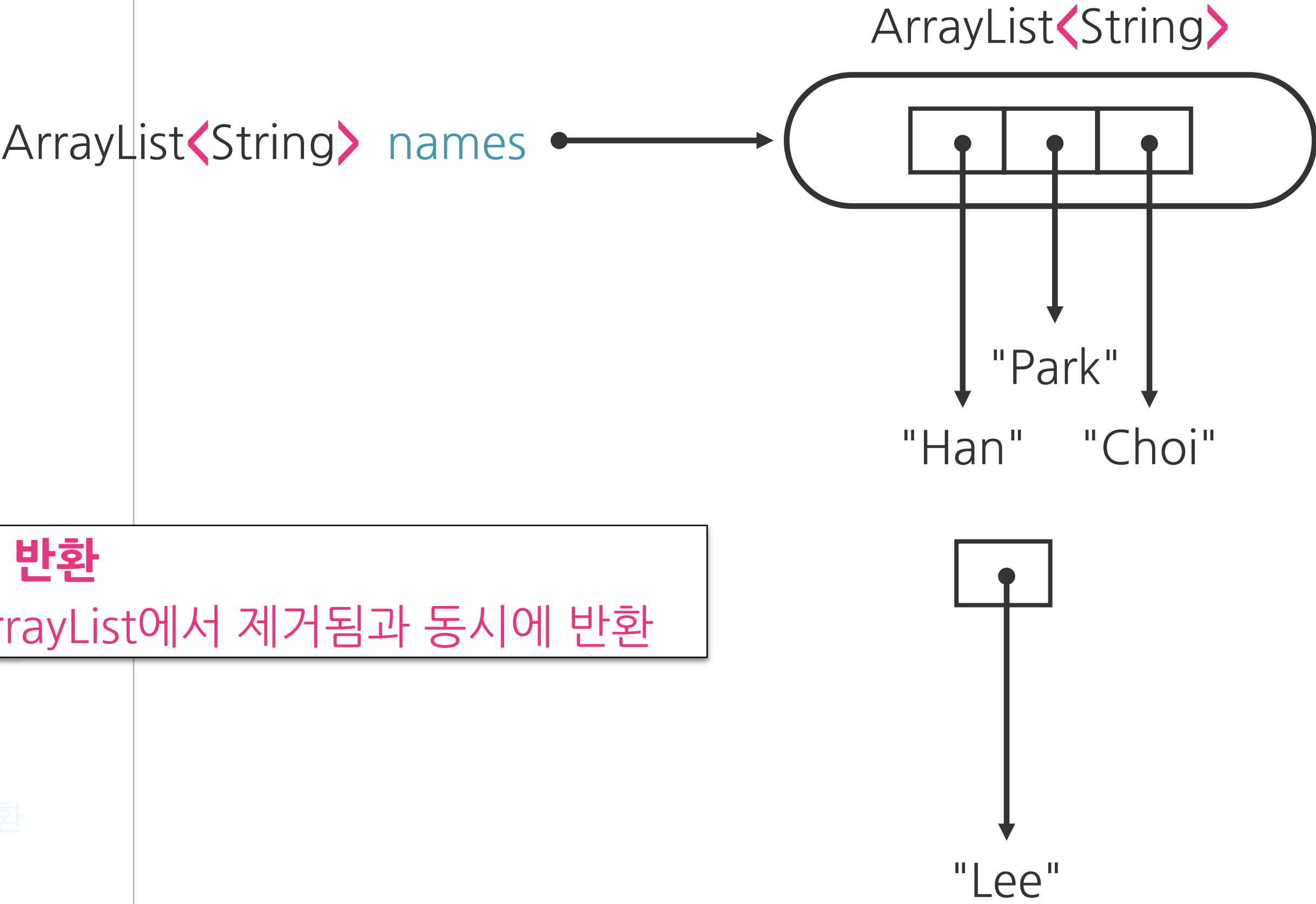
String 데이터 추가

0번 인덱스 데이터를 "Han"으로 변경

특정 인덱스 요소 삭제 및 반환
1번 인덱스의 "Lee"가 ArrayList에서 제거됨과 동시에 반환

i번 인덱스 값을 반환

remove() 메소드 수행 시
삭제 인덱스의 데이터는 한 칸씩 당겨져 빈자리를 채움



실행 결과

Han Park Choi

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기

String 저장을 위한
ArrayList 객체 생성

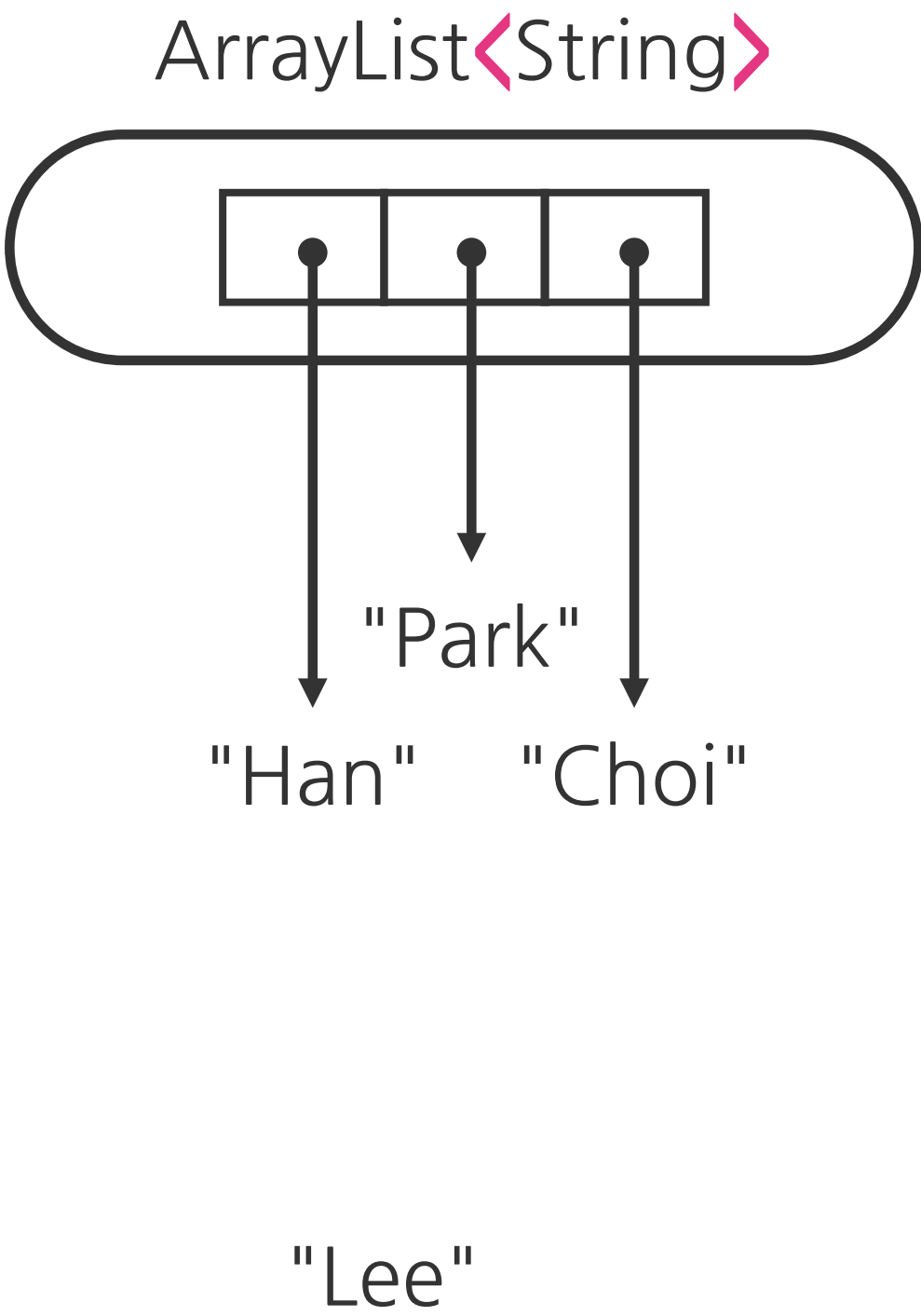
String 데이터 추가

0번 인덱스 데이터를 "Han"으로 변경

특정 인덱스 요소 삭제 및 반환
1번 인덱스의 "Lee"가 ArrayList에서 제거됨과 동시에 반환

i번 인덱스 값을 반환

ArrayList<String> names



실행 결과

Han Park Choi

실습 9-4

ArrayList 클래스 기본 활용

ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기

String 저장을 위한
ArrayList 객체 생성

String 데이터 추가

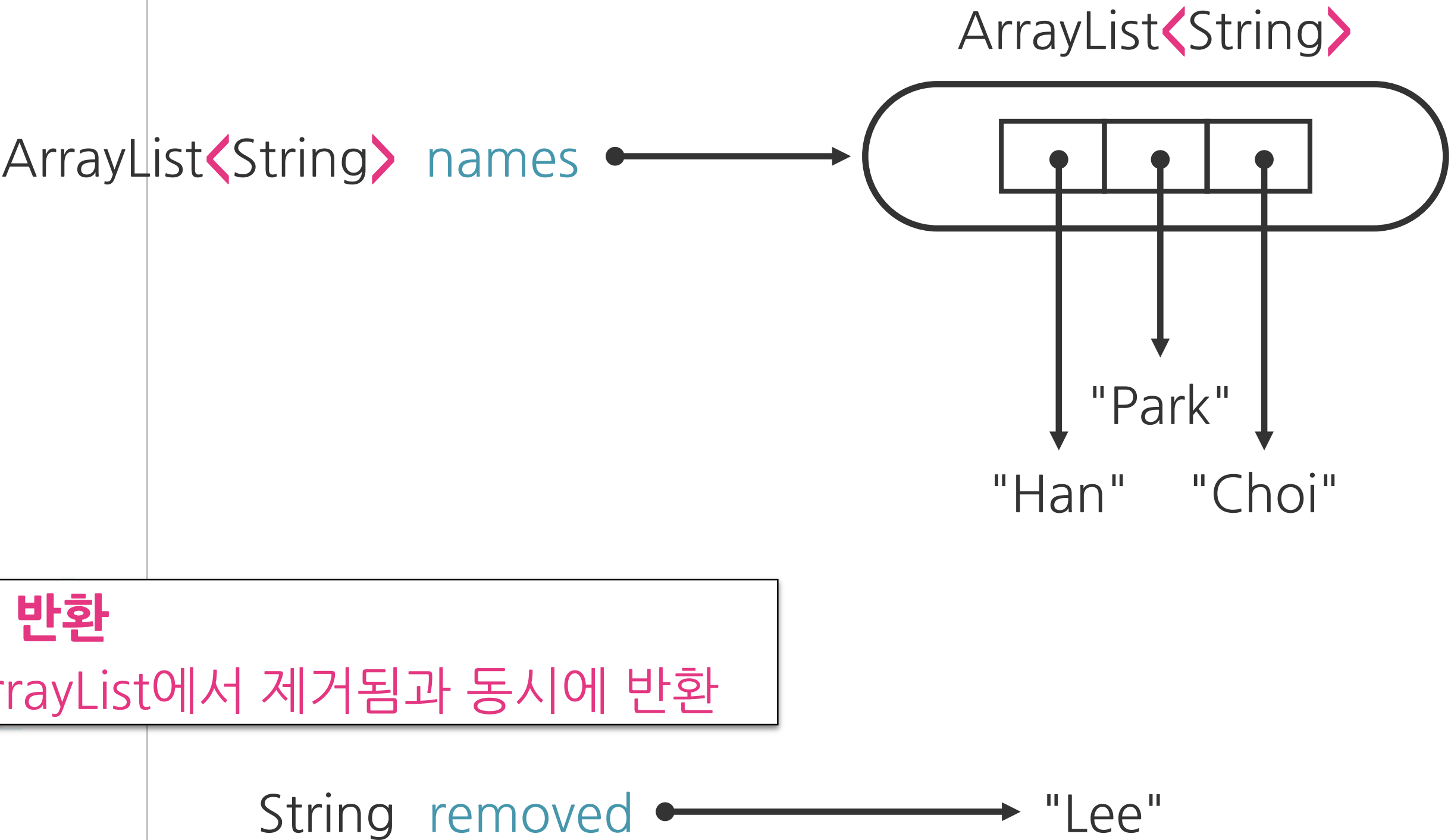
0번 인덱스 데이터를 "Han"으로 변경

1번 인덱스

ArrayList

i번 인덱스 값을 반환

특정 인덱스 요소 삭제 및 반환
1번 인덱스의 "Lee"가 ArrayList에서 제거됨과 동시에 반환




```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기

String 저장을 위한
ArrayList 객체 생성

String 데이터 추가

0번 인덱스 데이터를 "Han"으로 변경

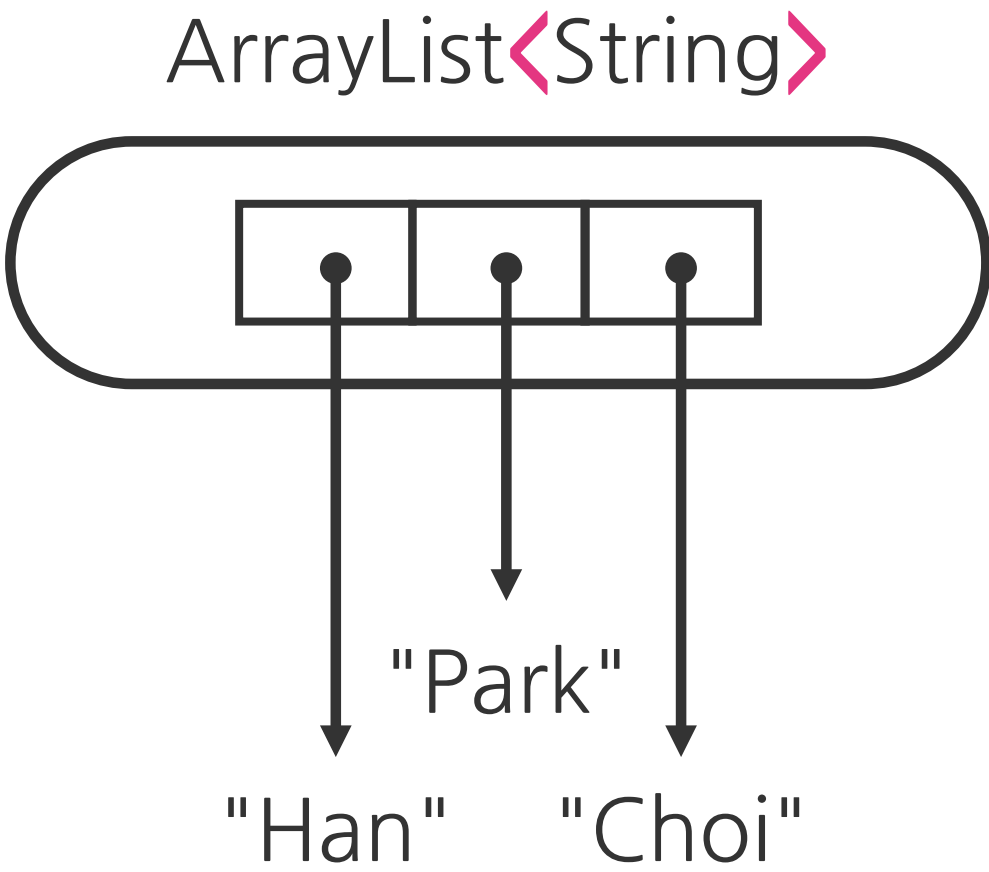
1번 인덱스 데이터 제거 후, 해당 값 반환

ArrayList에 담긴 데이터의 개수 반환

반복문을 통해 ArrayList를 순회 출력
저장된 요소의 개수는 **size()** 메소드를 통해,
특정 인덱스의 객체는 **get()** 메소드를 통해 가져올 수 있음.

ArrayList<String> names

String removed



"Lee"

실행 결과

Han Park Choi

실습 9-4

ArrayList 클래스 기본 활용

ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기

String 저장을 위한
ArrayList 객체 생성

String 데이터 추가

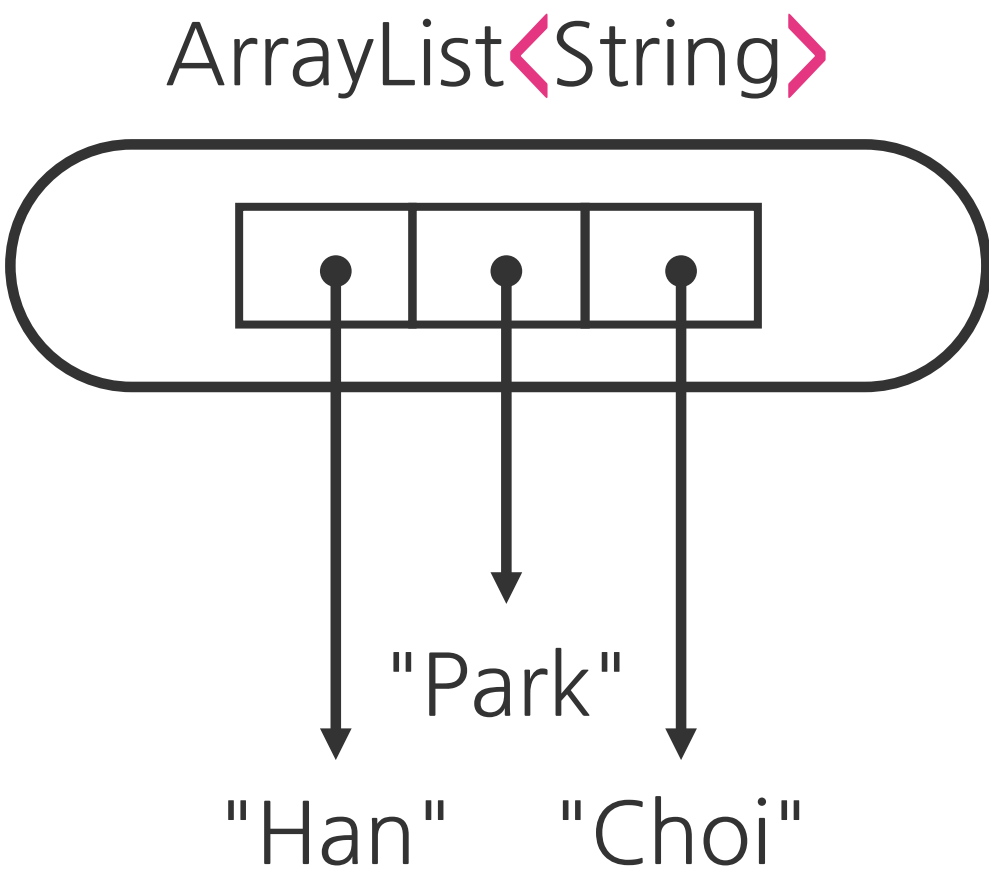
0번 인덱스 데이터를 "Han"으로 변경

1번 인덱스 데이터 제거 후, 해당 값 반환

ArrayList에 담긴 데이터의 개수 반환

반복문을 통해 ArrayList를 순회 출력
저장된 요소의 개수는 **size()** 메소드를 통해,
특정 인덱스의 객체는 **get()** 메소드를 통해 가져올 수 있음.

ArrayList<String> names



String removed

"Lee"

실행 결과

Han Park Choi

실습 9-4

ArrayList 클래스 기본 활용

ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기

String 저장을 위한
ArrayList 객체 생성

String 데이터 추가

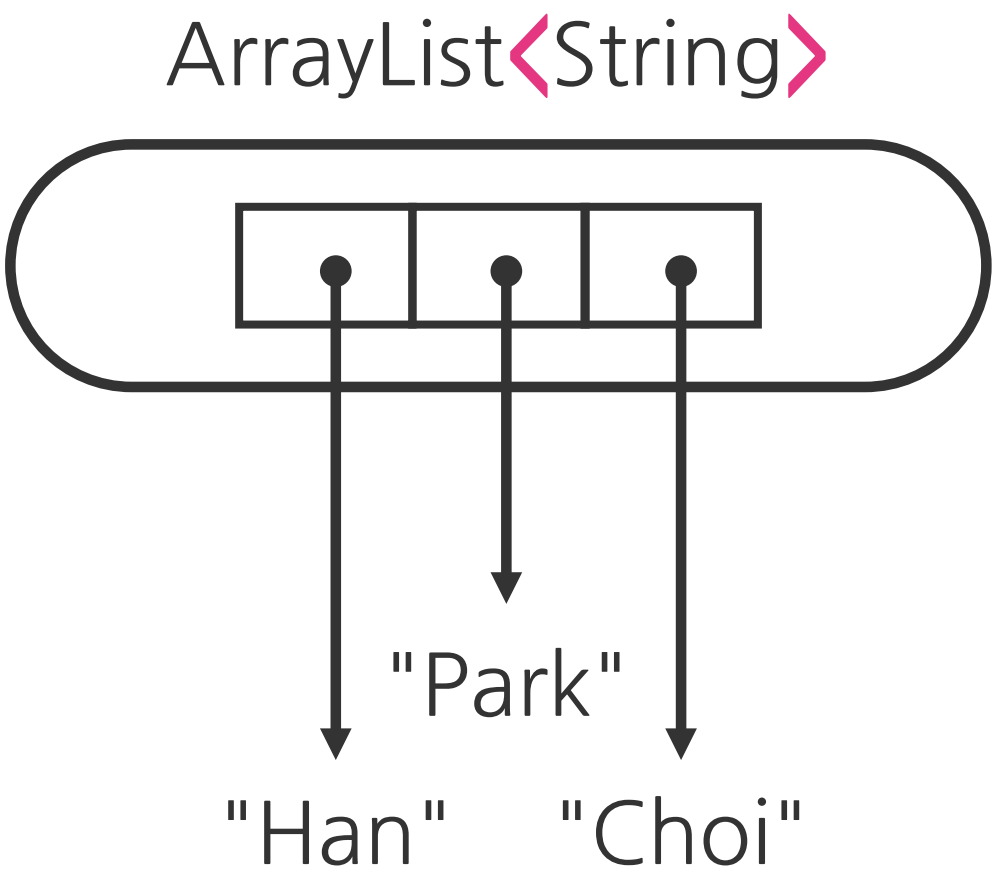
0번 인덱스 데이터를 "Han"으로 변경

1번 인덱스 데이터 제거 후, 해당 값 반환

ArrayList에 담긴 데이터의 개수 반환

반복문을 통해 ArrayList를 순회 출력
저장된 요소의 개수는 **size()** 메소드를 통해,
특정 인덱스의 객체는 **get()** 메소드를 통해 가져올 수 있음.

ArrayList<String> names



String removed

"Lee"

실행 결과

Han Park Choi

실습 9-4

ArrayList 클래스 기본 활용

ch09/ex9_4/ArrayListTest.java

```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05         ArrayList<String> names = new ArrayList<String>();
06
07         names.add("Kim");
08         names.add("Lee");
09         names.add("Park");
10         names.add("Choi");
11
12         names.set(0, "Han");
13         String removed = names.remove(1);
14
15         for (int i = 0; i < names.size(); i++) {
16             System.out.printf("%s ", names.get(i));
17         }
18     }
19 }
```

API 불러오기

String 저장을 위한
ArrayList 객체 생성

String 데이터 추가

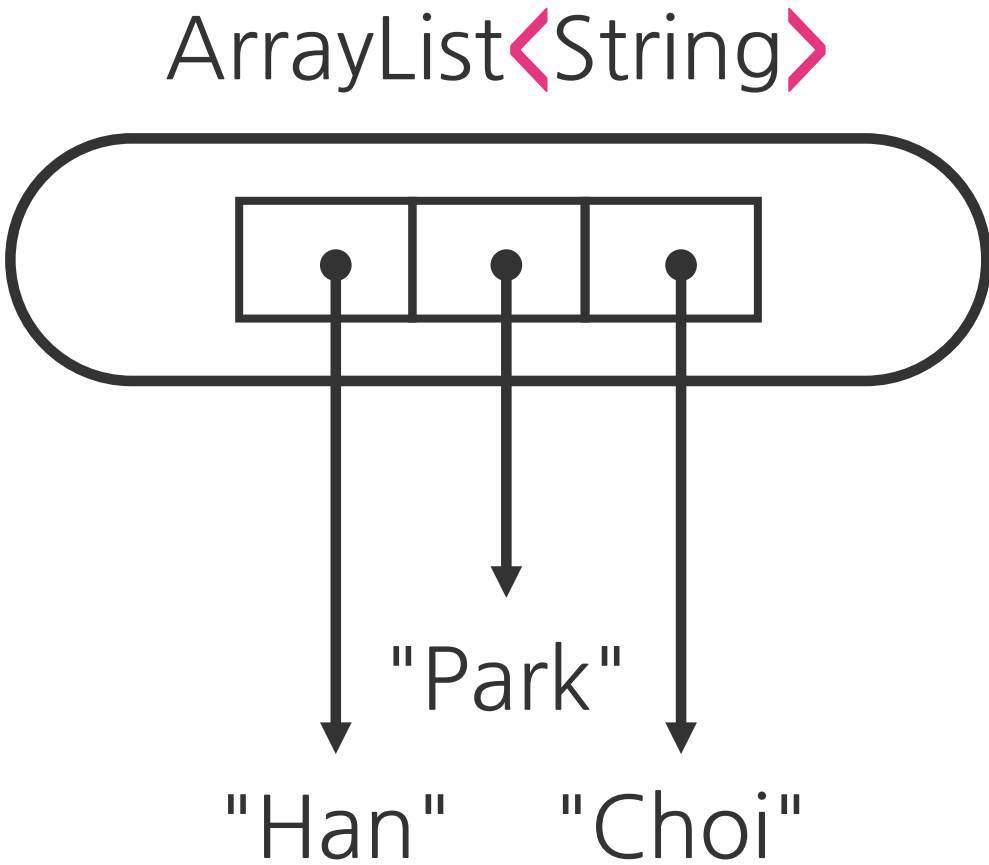
0번 인덱스 데이터를 "Han"으로 변경

1번 인덱스 데이터 제거 후, 해당 값 반환

ArrayList에 담긴 데이터의 개수 반환

반복문을 통해 ArrayList를 순회 출력
저장된 요소의 개수는 **size()** 메소드를 통해,
특정 인덱스의 객체는 **get()** 메소드를 통해 가져올 수 있음.

ArrayList<String> names



String removed

"Lee"

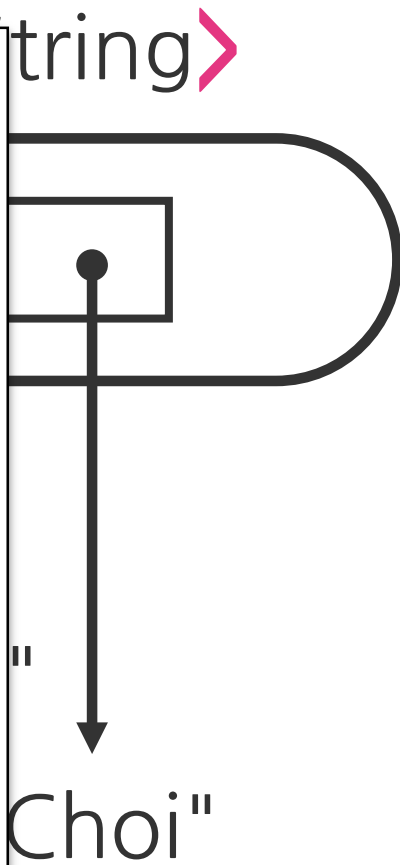
실행 결과

Han Park Choi


```
01 import java.util.ArrayList;
02
03 public class ArrayListTest {
04     public static void main(String[] args) {
05
06         ArrayList<String> list = new ArrayList<>();
07
08         list.add("Choi");
09
10         System.out.println(list);
11
12         // i번 인덱스 값을 반환
13         String name = list.get(0);
14
15         System.out.println(name);
16     }
17 }
18
19 }
```

표 9-3 ArrayList 클래스 설명 <https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/util/ArrayList.html>

코드	설명
ArrayList<E>()	E 타입 객체 관리를 위한 ArrayList 클래스의 생성자
void add(E element)	마지막 요소의 다음 인덱스에 element 객체를 추가하는 메소드
E set(int index, E element)	특정 index에 위치한 요소를 element로 변경하는 메소드
E remove(int index)	index 위치 요소를 제거와 동시에 반환하는 메소드
int size()	ArrayList에 담긴 요소의 개수를 반환하는 메소드



실행 결과

Han Park Choi

Quiz

**02 Math.random() 메소드와 Random 클래스의
nextDouble() 메소드의 차이점을 서술하시오.**

**03 Item 객체를 담기 위한 ArrayList 객체 생성 코드를
작성하시오.**

Thank You!