객체지향 프로그래밍 및 실습

4주차. 복잡한 Methods

1. 정적 필드, 정적 메서드

1. 정적 필드, 정적 메서드

- static 키워드
  - 정적(static) 멤버는 클래스(객체가 아님!)에 소속되어있다.
  - 쉽게 말하면 static 키워드가 붙은 변수, 메서드가 이에 해당

```java
package org.statickey;

public class Circle {
    private static final double PI = 3.14;
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    public double getArea() {
        return radius * radius * PI;
    }

    public static double getPI() {
        return PI;
    }
}
```

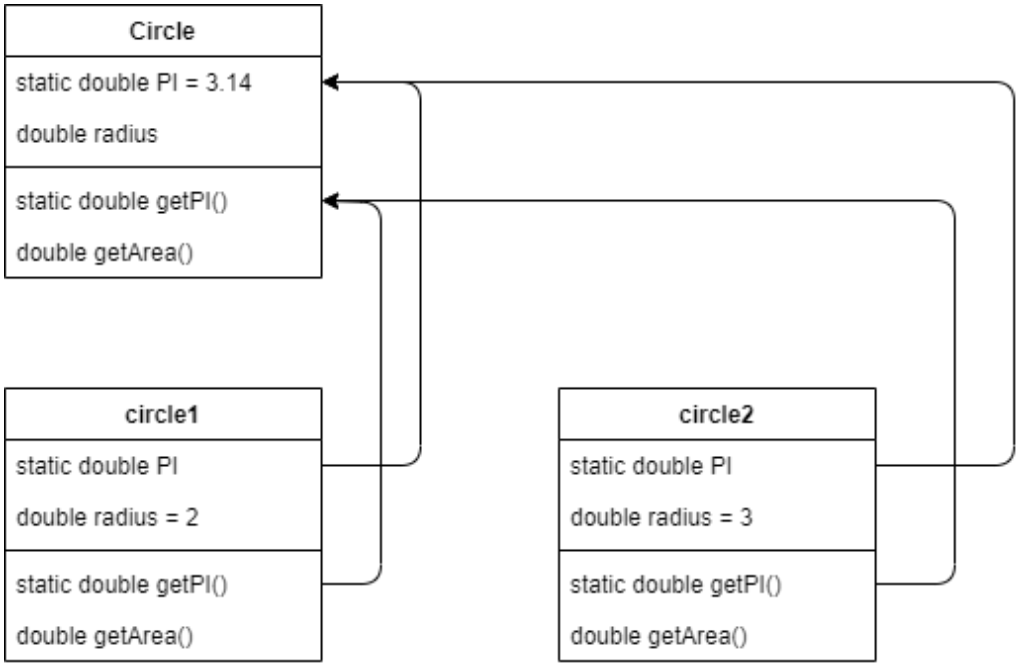

AJOU-MDC
국방디지털융합학과

1. 정적 필드, 정적 메서드

```java
package org.statickey;

public class CircleTest {
    public static void main(String[] args) {
        Circle.getArea();   // 에러발생: Cannot make a static reference to the non-static method getArea() from the type Circle

        Circle circle1 = new Circle(2);
        circle1.getArea();  // getArea() 메소드는 non-static 메소드이기 때문에 인스턴스를 먼저 생성한뒤, 객체를 통해 호출할 수 있다.

        Circle.getPI();     // getPI() 메소드는 static 메소드이기 때문에 클래스를 통해 호출할 수 있다.
    }
}
```

1. 정적 필드, 정적 메서드

- static 키워드
  - Circle circle1 = new Circle(2);
  - Circle circle2 = new Circle(3);



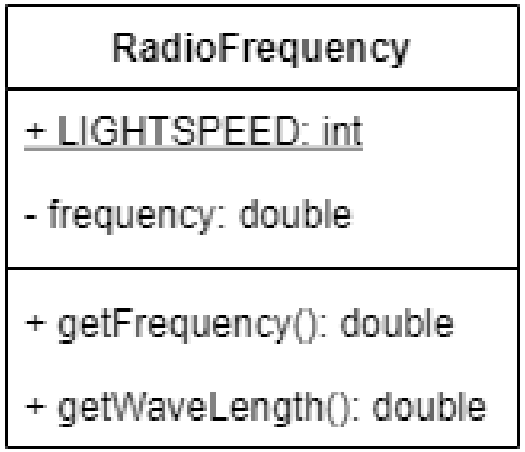
1. 정적 필드, 정적 메서드

- Static to non-static 접근
  - Static 에서 Non-static ✗
  - Static에서 Static ○
  - Non-static에서 static ○
  - Non-static에서 Non-static ○

```java
package org.statickey;

public class Circle {
    private static final double PI = 3.14;
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    public double getArea() {
        return radius * radius * PI;
    }

    public static double getPI() {
        return PI;
    }

    public static void accessGetArea() {
        getPI();    // 정상적으로 작동
        getArea();  // 에러 발생 : Cannot make a static reference to the non-static method getArea() from the type Circle

        double a = radius;  // 에러 발생
        double b = PI;      // 정상적으로 작동
    }

    public void nonStaticAccess() {
        getPI();
        getArea();  // 둘 다 정상적으로 작동

        double a = radius;  // 에러 발생
        double b = PI;      // 정상적으로 작동
    }
}

```

1. 정적 필드, 정적 메서드

- **실습 문제 1**
  - RadioFrequency 클래스를 조건에 맞게 작성 하시오.
  - RadioFrequencyTest 클래스에 main 메소드를 생성하고 30MHz 주파수의 파장을 출력하시오.



파장 = 광속 / 주파수

2. 인자 Promotion 및 Casting

2. 인자 Promotion 및 Casting

- Promotion Rules
  - Math.sqrt(double x)는 x의 제곱근을 구하는 메소드
  - System.out.println(Math.sqrt(4));
  - Int 타입의 4를 넣었는데 잘 작동이 된다!
  - 이는 int -> double이 Java Promotion rules를 잘 만족하기 때문

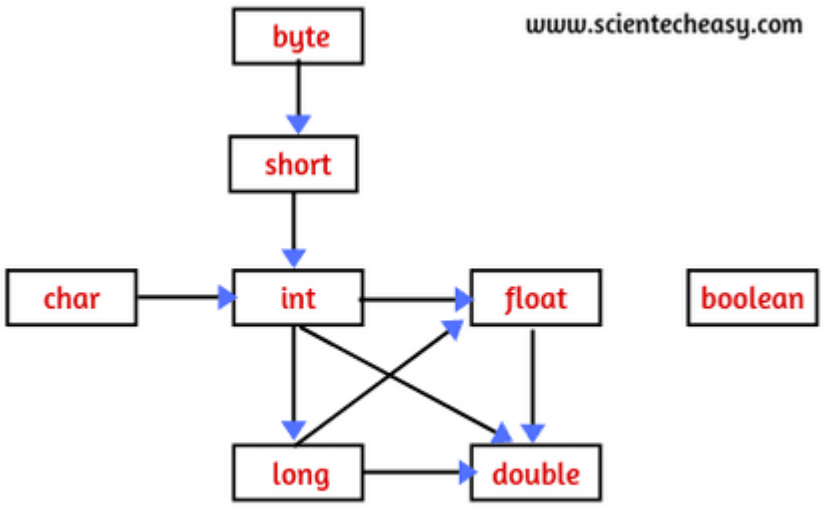


Fig: Automatic type conversion that Java allows.

```java
public class PromotionTest {
    public static void main(String[] args) {
        intParameter(5);
        intParameter(5.0);  // error

        doubleParameter(5);
        doubleParameter(5.0);
    }

    public static void intParameter(int a) {

    }

    public static void doubleParameter(double a) {

    }
}
```

3. 메서드 오버로딩

3. 메서드 오버로딩

- 같은 이름의 메서드 - 다른 인자

```java
public class OverloadingTest {
    public static void main(String[] args) {
        print(1);
        print(1, 4);
        print(5.4, 7);
    }

    public static void print(int a) {
        System.out.printf("%d\n", a);
    }

    public static void print(int a, int b) {
        System.out.printf("%d %d\n", a, b);
    }

    public static void print(double a, int b) {
        System.out.printf("%f %d\n", a, b);
    }
}
```

```java
public class OverloadingTest {
    public static void main(String[] args) {
        System.out.println(getData(5));
        System.out.println(getData(1, 9))
    }

    public static String getData(int a) {
        return String.format("%d", a);
    }

    public static int getData(int a, int b) {
        return a + b;
    }
}
```

3. 메서드 오버로딩

- **실습 문제 2**
  - MethodOverloadingTest 클래스에 main 메서드를 생성하고 아래 사항을 구현해 보시오.

  - 주어진 매개변수를 모두 합쳐서 반환하는 sum() 메서드가 오버로딩되도록한다.
  - sum(int a, int b): int
  - sum(int a, int b, int c): int
  - sum(double a, double b): double
  - sum(double a, double b, double c): double
  - sum(List<Integer> list): int
  - sum(List<Double> list): double

4. Arrays

4. Arrays

- java.util.Arrays 클래스
  - TCPSchool 내용 발췌

대표적인 Arrays 메소드

Arrays 클래스의 메소드는 매우 다양하며, 그중에서 많이 사용되는 메소드는 다음과 같습니다.

메소드	설명
static <T> List<T> asList(T... a)	전달받은 배열을 고정 크기의 리스트(list)로 변환하여 반환함.
static int binarySearch(Object[] a, Object key)	전달받은 배열에서 특정 객체를 이진 검색 알고리즘을 사용하여 검색한 후, 그 위치를 반환함.
static <T> T[] copyOf(T[] original, int newLength)	전달받은 배열을 특정 길이의 새로운 배열로 복사하여 반환함.
static <T> T[] copyOfRange(T[] original, int from, int to)	전달받은 배열의 특정 범위에 해당하는 요소만을 새로운 배열로 복사하여 반환함.
static boolean equals(Object[] a, Object[] a2)	전달받은 두 배열이 같은지를 확인함.
static void fill(Object[] a, Object val)	전달받은 배열의 모든 요소를 특정 값으로 초기화함.
static void sort(Object[] a)	전달받은 배열의 모든 요소를 오름차순으로 정렬함.

4. Arrays

- asList()
  - 배열 타입을 List 타입으로 변환

```java
String[] strArray = { "HI", "Hello", "World", "Whatup", "GoodDay", "NiceDay" };

List<String> strList = Arrays.asList(strArray);

for (String str : strList) {
    System.out.println(str);
}
```

4. Arrays

- fill()
  - 배열의 모든 요소를 주어진 값으로 할당

```java
int[] intArray = { 5, 3, 4, 8, 7, 9, 2 };

for (int element : intArray) {
    System.out.printf("%s ", element);
}
System.out.println("\n");

Arrays.fill(intArray, 50);

for (int element : intArray) {
    System.out.printf("%s ", element);
}
```

4. Arrays

- sort()
  - 배열을 오름차순으로 정렬

```java
int[] intArray = { 5, 1, 9, 7, 0, 3, 5, 4, 3, 8 };

for (int element : intArray) {
    System.out.printf("%d ", element);
}

System.out.println("\n");

Arrays.sort(intArray);

for (int element : intArray) {
    System.out.printf("%d ", element);
}
```

4. Arrays

- binarySearch()
  - 주어진 데이터가 배열의 어느 위치에 있는지 탐사

```java
Random rand = new Random(System.currentTimeMillis());
int[] intArray = rand.ints(100).map(randNum -> randNum % 200 + 200).toArray();
// 위의 코드는 1~200 까지의 난수를 100개 길이로 가지는 배열을 생성한다
// 아직 배우지 않은 내용이므로, 역할만 알고 있으면 됨


Arrays.sort(intArray);

int index = 35;
int indexElement = intArray[index];

int foundIndex = Arrays.binarySearch(intArray, indexElement);

System.out.println(index == foundIndex);
```

5. 실습 종합 문제

- **실습 문제 3**
  - 아래 UML 다이어그램을 준수하여 프로그램을 개발 하시오.
  - isOverlap(Circle2D a, Circle2D b) 메서드는 두 원 a와 b가 겹치는지 아닌지 여부를 반환
  - PI 클래스 변수에는 3.14를 입력