Wrapper classes:

Each primitive type has a corresponding reference defined in java.lang:

- int
- Integer
- long
- Long
- short
- Short
- byte
- Byte
- char
- Character
- float
- Float
- double
- Double
- boolean
- Boolean
Each reference type contains just one data field of the corresponding primitive type.

Ex.

```java
Integer A = new Integer(6);
```

creates

A wrapper class also contains methods and constants for things.
like String to int & int to String conversion.

Methods in Integer:

- equals()
- toString()
- valueOf()
- compareTo()
- parseInt()

Constants:
- MAX_VALUE
- MIN_VALUE
Note:

Shortcut for

```
Integer A = new Integer(6);
```

is:

```
Integer A = 6;
```

Similar to:

```
String s = new String("happy");
```

vs.

```
String s = "happy";
```
Ex. IntegerTest.java

```java
a += 10;
```

Same as

```java
a = a + 10;
```

These become garbage and will be recycled by the garbage collector.
Google: 2's complement representation of integers.

Exercise: Write test
programs for other wrapper
classes. (Try Long and Boolean)
Rational numbers:

Rational numbers are fractions

\[ \frac{a}{b} \] where \( a, b \) are integers and \( b \neq 0 \).

Ex. Rational 1/2

Note:

\[ \frac{-2}{-4} = \frac{-2}{4} = \frac{-1}{2} \]
Compare to (1):

\[ \frac{a}{b} < \frac{c}{d} = \Rightarrow \]

\[ \Leftrightarrow ad < bc = \Rightarrow \]

\[ \Leftrightarrow ad - bc < 0 = \Rightarrow \]

\[ \text{return: } \quad -1 \quad 0 \quad +1 \]

\[ \text{add(1)}: \]

\[ \frac{a}{b} + \frac{c}{d} = \frac{ad + bc}{bd} \]

\[ \text{sub(1)}: \]

\[ \frac{a}{b} - \frac{c}{d} = \frac{ad - bc}{bd} \]
Ex. Vector.java

Pa6:
Create Complex.java

\[ a + bi \]

\[ b \]

\[ a \]

\[ i \] is the "imaginary" unit.

\[ i^2 = -1 \]