Junit

Junitとは

- Java言語で開発されたプログラムの単体テスト（ユニットテスト）を行なうためのソフトウェア。
- Junitを使うことで作成したJavaクラスやメソッドの単体テストコードが記載できるようになる。
- Junitには単体テストに当たって活用できるメソッドやアノテーションが複数提供されている。

基本的なテストクラス

基本的なテストクラス（テストコード）は以下のように作成する

```
public class BasicTest {
    public BasicTest() {
    }
    @BeforeAll
    public static void setUpClass() {
    }
    @AfterAll
    public static void tearDownClass() {
    }
    @BeforeEach
    public void setUp() {
    }
    @AfterEach
    public void tearDown() {
    }
    @Test
    @DisplayName("Comment")
    public void testSomeMethod() {
    }
}
```

アノテーション

上記で使用されているアノテーションを以下に解説する。

- @BeforeAll
  - テストクラスが実行されて、初めに必ず実行するメソッド
- @AfterAll
  - テストクラスが実行されて、最後に必ず実行するメソッド
- @BeforeEach
  - 後述の@Testアノテーションが付与された各テストメソッドの前に、必ず実行するメソッド
  - Junit4では@Beforeに該当
- @AfterEach
  - 後述の@Testアノテーションが付与された各テストメソッドの後に、必ず実行するメソッド
  - Junit4では@Afterに該当

- @Test
  - このアノテーションが付与されたメソッドはテストメソッドとなり、各テストケースごとに作成する。
- @DisplayName
  - JUnitテストクラス・メソッドの表示名

アサーション

> Javaプログラムの実行結果などを比較することができる仕組み

アサーションメソッドとして、Junitでは以下のようなメソッドが提供されている。

- assertEquals(expected, actual)
  - expectedに期待値、actualに実測値を記載し、期待値と一致しているかを検証する。
  - 等しくない場合、アサーションエラーが発生してテストが失敗する。
- assertTrue(bool)
  - 引数に指定した値が true/falseのどちらであるかを確認するアサーション。
  - 引数にはboolean血を設定する。
- assertThat(actual, expected)
  - asertEqualsと引数の順序が逆になっているので注意
  - Hamcrestというアサーションライブラリのメソッドを第二引数のexpectedに入れて使用する事で、多彩なアサーションが可能

使用例

```
// 文字列比較
String actualString = "Hello, World!";
assertThat(actualString, containsString("Hello")); // actualStringに"Hello"を含むか検証
assertThat(actualString, startsWith("Hello")); // actualStringが"Hello"で始まるか検証
assertThat(actualString, endsWith("World!")); // actualStringが"World!"で終わるか検証

// コレクション比較
List<String> actualList = Arrays.asList("apple", "banana", "orange");
assertThat(actualList, hasSize(3)); // actualListのサイズが3か検証
assertThat(actualList, contains("apple", "banana", "orange")); // actualListが指定した順序で要素を含むか検証
assertThat(actualList, containsInAnyOrder("banana", "orange", "apple")); // actualListが指定した順序を無視して要素を含むか検証

// オブジェクトのプロパティ（フィールド）比較
class Person {
    private String name;
    private int age;

    // コンストラクタ、ゲッター、セッターなど
}
```

```
Person actualPerson = new Person("Natsu", 25);
assertThat(actualPerson, hasProperty("name", equalTo("Natsu"))); // actualPersonの
nameプロパティが"Natsu"か検証
assertThat(actualPerson, hasProperty("age", greaterThan(20))); // actualPersonの
ageプロパティが20より大きいか検証
```