

Мамонов Антон ЗИСИП-19-1

1. Написать класс по расчету индекса массы тела ИМТ (см. аналитику в отдельном файле) в папке test.

2. Создать два тестовых сценария.

- Позитивный сценарий:

- Негативный сценарий:

Код программы:

```
private String getIMTResult(Float heightCm, Float weightKg) {
    var userIndex :long = Math.round(weightKg / Math.pow((heightCm / 100), 2));
    String userResult = null;

    if ((weightKg.equals(0.0f)) && (heightCm.equals(0.0f))) {
        userResult = "указаны некорректные рост и вес";
        return userResult;
    } else if (weightKg.equals(0.0f) | weightKg > (300f)) {
        userResult = "указан некорректный вес";
        return userResult;
    } else if (heightCm.equals(0.0f) | heightCm > (300f)) {
        userResult = "указан некорректный рост";
        return userResult;
    }

    if (userIndex <= 16) {
        userResult = "выраженному дефициту массы тела";
    } else if (userIndex > 16 && userIndex < 19) {
        userResult = "недостаточной массе тела";
    } else if (userIndex > 19 && userIndex < 25) {
        userResult = "нормальной массе тела";
    } else if (userIndex > 25) {
        userResult = "избыточной массе тела";
    }

    return userResult;
}
```

После запуска получаем
результат тестирования

✓ ✗ Main (com.company)	32 ms
✗ testZeroValues	19 ms
✗ testMaxHeight	3 ms
✗ testMaxWeight	6 ms
✗ testZeroWeght	3 ms
✓ testNormalWeight	1 ms
✓ testZeroHeight	0 ms

Тесты с позитивным сценарием:

Вводим рост и вес в пределах нормы, получаем ожидаемый результат

```
@Test
public void testNormalWeight() {
    var actualResult :String = getIMTResult( heightCm: 180f, weightKg: 75f);
    Assert.assertEquals( expected: "нормальной массе тела", actualResult);
}
```

Вводим рост = 0, получаем информацию о некорректном росте

```
@Test
public void testZeroHeight() {
    var actualResult :String = getIMTResult( heightCm: 0f, weightKg: 65f);
    Assert.assertEquals( expected: "указан некорректный рост", actualResult);
}
```

Тесты с негативным сценарием:

Вводим вес = 0, получаем неверный ответ

```
@Test
public void testZeroWeght() {
    var actualResult :String = getIMTResult( heightCm: 180f, weightKg: 0f);
    Assert.assertEquals( expected: "указан некорректный вес", actualResult);
}
```

```
org.junit.ComparisonFailure:
Expected :указан некорректный вес
Actual   :выраженному дефициту массы тела
<Click to see difference>
```

Вводим рост = 0 и вес = 0

```
@Test
public void testZeroValues() {
    var actualResult :String = getIMTResult( heightCm: 0f, weightKg: 0f);
    Assert.assertEquals( expected: "указаны некорректные рост и вес", actualResult);
}
```

Получаем информацию только о некорректном росте

```
org.junit.ComparisonFailure:
Expected :указаны некорректные рост и вес
Actual   :указан некорректный рост
<Click to see difference>
```

Вводим невозможный рост

```
@Test
public void testMaxHeight() {
    var actualResult :String = getIMTResult( heightCm: 305f, weightKg: 80f);
    Assert.assertEquals( expected: "указан некорректный рост", actualResult);
}
```

Получаем неверный ответ

```
org.junit.ComparisonFailure:
Expected :указан некорректный рост
Actual   :выраженному дефициту массы тела
<Click to see difference>
```

Вводим невозможный вес

```
@Test
public void testMaxWeight() {
    var actualResult :String = getIMTResult( heightCm: 180f, weightKg: 400f);
    Assert.assertEquals( expected: "указан некорректный вес", actualResult);
}
```

Получаем неверный ответ

```
org.junit.ComparisonFailure:
Expected :указан некорректный вес
Actual   :избыточной массе тела
<Click to see difference>
```