



Евгений Гродно

36 уровень



30 апреля 2018 18:16



1066



5

JUnit part II

Пост из группы Random

995223 участников

продолжение

начало тут -> [JUnit part I](#)



Еще хочу вам показать, как работает метод `Assert.fail(String)` – если будет вызван этот метод, тест будет провален. Удобно в случае если другие методы класса `Assert` не могут сделать нам проверку. Допустим нам нужно запретить принимать `new String("")`. Попробуем это сделать на конструкторе запретим передавать `name = "" || null, age = 0; Sex = null`. Поехали ...

Я добавил дополнительные поля для тестирования

```
1 private User user;  
2 private User user1;  
3 private User user2;  
4  
5 private User userNotAdd;  
6 private User userNotAdd1;
```

и изменил метод `setUp()`

```
1 @Before  
2 public void setUp() throws Exception {  
3     user = new User("Евгений", 35, Sex.MALE);  
4     user1 = new User("Марина", 34, Sex.FEMALE);  
5     user2 = new User("Алина", 7, Sex.FEMALE);  
6  
7     userNotAdd = new User("", 0, null);  
8     userNotAdd1 = new User(null, 0, null);  
9 }
```

и добавил три метода теста

```
1 @Test  
2 public void newUser_EMPTY_NAME() {  
3     for (User user : User.getAllUsers()){  
4         if (user.getName() != null && user.getName().isEmpty()) {  
5             Assert.fail("Попытка создания пользователя с пустым именем");  
6         }  
7     }  
8 }  
9  
10 @Test  
11 public void newUser_AGE_ZERO() {  
12     for (User user : User.getAllUsers()) {  
13         if (user.getAge() <= 0) {  
14             Assert.fail("Попытка создания пользователя с не допустимым возрастом");  
15         }  
16     }  
17 }  
18  
19 @Test  
20 public void newUser_SEX_NO_NULL() {  
21     for (User user : User.getAllUsers()) {
```

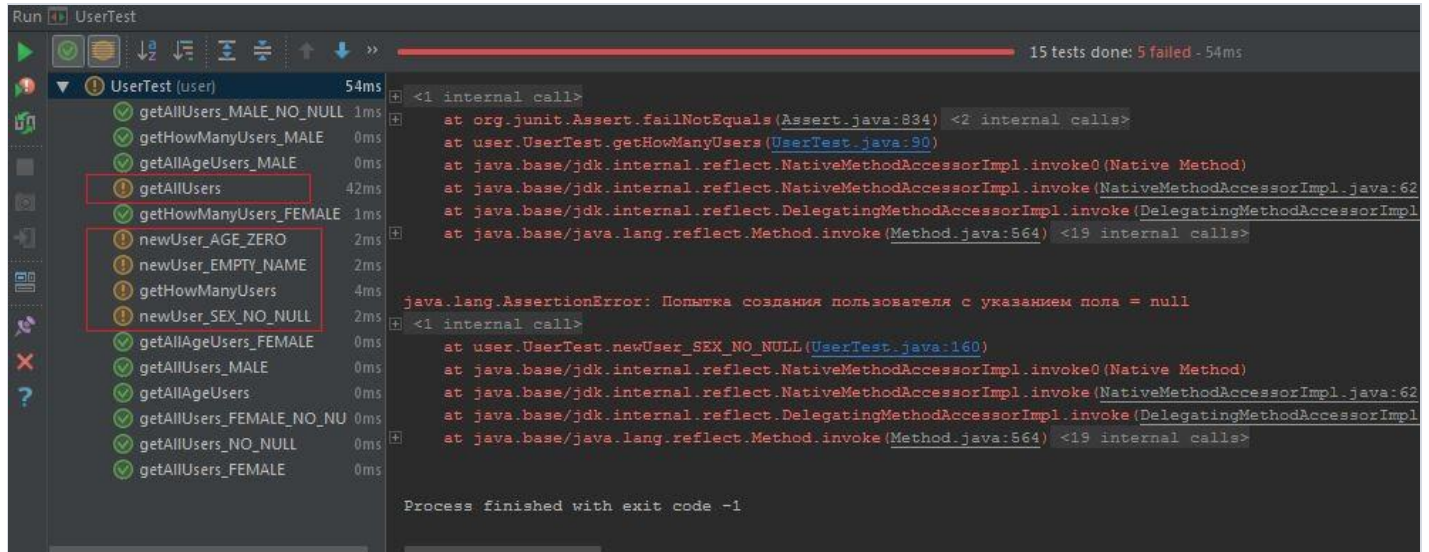
```

22         if (user.getSex() == null) {
23             Assert.fail("Попытка создания пользователя с указанием пола = null");
24         }
25     }
26 }

```

В тестах мы проходим по списку добавленных пользователей, и смотрим есть ли не допустимые данные.

Запускаем тесты оптом, и видим такую картину...



Тут видно, наши тесты для конструктора поломались, а так же зацепило еще пару тестов

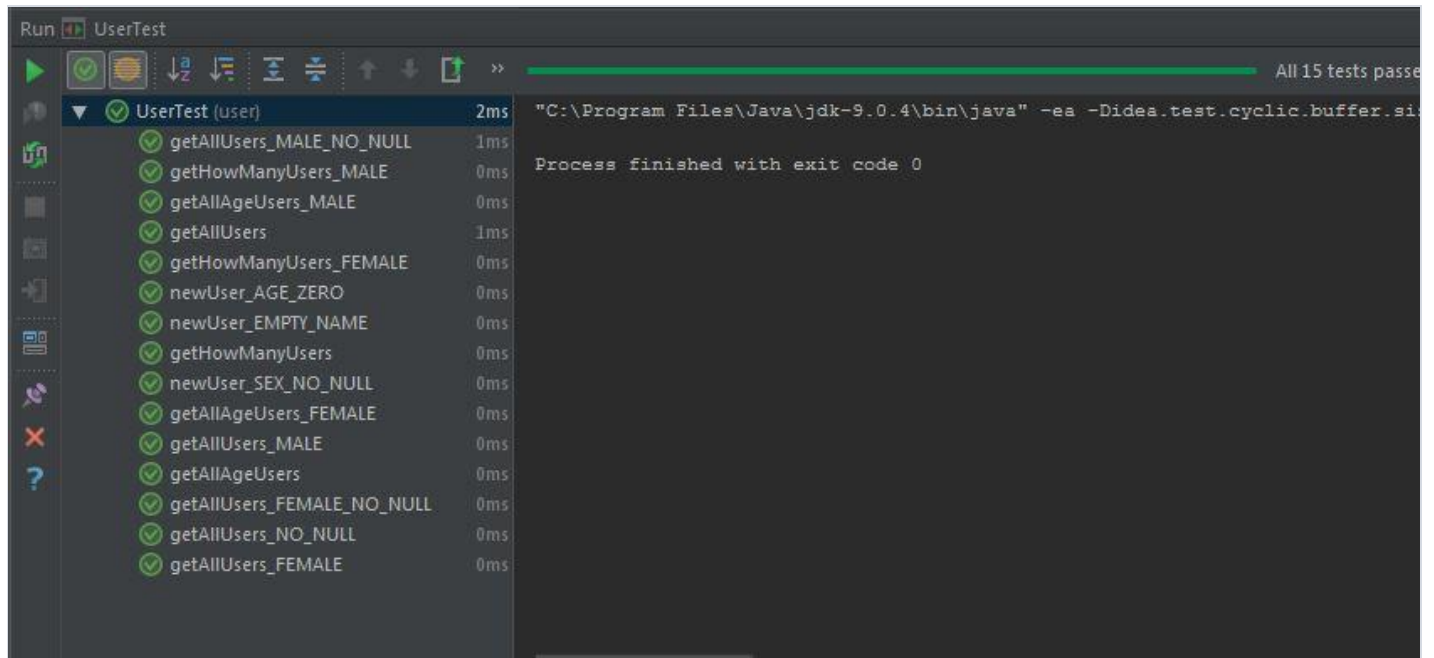
Теперь нам нужно изменить конструктор, чтобы он не добавлял в список пользователей, пользователя с недопустимыми параметрами, сделаем это так:

```

1  public User(String name, int age, Sex sex) {
2      if (name != null && !name.isEmpty() && age > 0 && sex != null){
3          this.name = name;
4          this.age = age;
5          this.sex = sex;
6
7          if (!hasUser()) {
8              countId++;
9              this.id = countId;
10             allUsers.put(id, this);
11         }
12     }
13 }

```

Запускаем наши тесты и видим, что у нас все по красоте



Подведем итог, **JUnit** отличный инструмент для покрытия своего кода тестами. **JUnit** работает еще лучше, в связке с **Maven**, при сборке проекта, **Maven** обязательно проведет все тесты, и потом соберет наш готовый проект, но если тесты будут провалены, проект не будет собран, и мы будем знать причины, и так будет легче разобраться где именно поломка.

Второй плюс **JUnit** это случай **Рефакторинга**, тесты **JUnit** помогут нам проследить чтобы логика нашей программы не поломалась. А в больших проектах без системы автоматического тестирования жизнь не возможна, хотя может и возможна, но двигаться ваш проект будет со скоростью черепахи ... или как то так ...



Если мы хорошо покроем тестами наш код, то можем быть уверены, что у нашего клиента, наш продукт будет отлично работать.

P/S Выражаю огромную благодарность Сергееву Виктору!!!

Вот ссылки на видео урок, в котором он нас обучал:

- [Тестирование java](#)
- [Тестирование java \(продолжение\)](#)

Спасибо за внимание!!!

Хорошего вам настроения!!!

И да спаси вас Бог от индускода!!!

...еще если у кого будет интерес покопаться в исходном коде для статьи... стучитесь... с поделюсь



Комментарии (5)

популярные **новые** старые

ДЛЯ ТОГО, ЧТО БЫ ОСТАВИТЬ КОММЕНТАРИЙ ВЫ ДОЛЖНЫ [АВТОРИЗОВАТЬСЯ](#)



Ваня Жуков 40 уровень, Санкт-Петербург
4 мая, 00:22

```
1  @Test
2  public void newUser_EMPTY_NAME() {
3      for (User user : User.getAllUsers()){
4          if (user.getName().isEmpty() && user.getName() != null) { // По иде
5              Assert.fail("Попытка создания пользователя с пустым именем");
6          }
7      }
8  }
```



Евгений Гродно 36 уровень
4 мая, 01:06

Спасибо! Подправил



Моряк Папай 40 уровень
1 мая, 08:57

Суперская книга: [Practical Unit Testing with JUnit and Mockito](#)

Подоедет даже людям со слабым английским, заодно практика чтения будет. Автор пишет простым, не академическим языком, вставляет шутки и т.д. Рекомендую всем кому интересно покопаться и досконально разобраться с юнит тестированием.



LexSav 35 уровень
1 мая, 08:15

Осталось еще про параметризованные тесты,
и азы JUnit ты покрыл)

0



Nikita Koliadin 40 уровень, Днепр
1 мая, 08:07

Молоток! Думаю многие узнаю новое для себя!
Виктору также спасибо!

0



Программистами не рождаются
© 2018