1. Sintaksės klaidos

UnitTest klasėje buvo įsivėlusi sintaksės klaida *(pav 1):*

```
@Test
void testPhonePrefix_true){
void testPhonePrefix_true(){
    assertTrue(PhoneValidator.validatePrefix("LT", "+37000000000"));
}
```

pav. 1 sintaksės klaida.

Taip pat teko pervadyti vieną UnitTest metodą testPhonePrefix_true(), kuris neleido kompiliuotis kodui, nes turėjo identišką vardą su kitu UnitTest klasės metodu. *(pav2 ir pav3)*

BEFORE:

```
53      @Test
54      void testPhonePrefix_true(){
55          assertTrue(PhoneValidator.validatePrefix( lt: "LT",  s: "800000000"));
56      }
57      @Test
58      void testPhonePrefix_true(){
59          assertTrue(PhoneValidator.validatePrefix( lt: "LT",  s: "+37000000000"));
60      }
61      @Test
62      void testPhonePrefix_false() { assertFalse(PhoneValidator.validatePrefix( lt: "LT",  s: "900000000")); }
```

pav. 2 prieš metodo pervadinimą.

AFTER:

```
53      @Test
54      void testPhonePrefixLocal_true(){
55          assertTrue(PhoneValidator.validatePrefix( lt: "LT",  s: "800000000"));
56      }
57      @Test
58      void testPhonePrefix_true() { assertTrue(PhoneValidator.validatePrefix( lt: "LT",  s: "+37000000000")); }
61      @Test
62      void testPhonePrefix_false() { assertFalse(PhoneValidator.validatePrefix( lt: "LT",  s: "900000000")); }
```

pav. 3 po metodo pervadinimo

2. Kiti trūkumai:

- Nebuvo sukurta testų, kurie patikrintų validavimo metodus su null reikšmėmis.
- Testuojami tik pavieniai metodai, t,y., nėra vieno validate metodo, kuris apjungtų visus metodus, testuojančius mažas specifines dalis, ir patikrintų, ar nusiųsti duomenys yra korektiški. Vadinasi, bibliotekos naudotojas turi naudotis keliais metodais norėdamas validuoti norimas reikšmes.
- Specialusis simbolis negali būti pirma ar paskutinė adreso raidė. VISGI šie testai prieštarauja vienas kitam:

```
@ParameterizedTest
@ValueSource(strings = {"TesT@email.com", "te123st@email.com", "!#$%&'*+-/=?^_`{|}~@email.com"})
void testEmailSymbols_true(String email) { assertTrue(EmailValidator.validateSymbols(email)); }
```

pav. 4 metodas testEmailSymbols_true()

```
@ParameterizedTest
@ValueSource(strings = {".test@email.com", "test.@email.com", "te..st@email.com"})
void testEmailDotPlacement_false(String email) { assertFalse(EmailValidator.validateSymbols(email)); }
```

pav. 5 metodas testEmailDotPlacement_false()

testEmailSymbols_true() *(pav 4)* metode, `"!#$%&'*+-/=?^_`{|}~@email.com"` adresas yra validus, nors jis prasideda specialiu simboliu ir juo pasibaigia, nekalbant, kad specialus simbolis negali pasikartoti du kartus iš eilės tarp jų nesant raidėms ar skaičiams. TAČIAU kitame teste testEmailDotPlacement_false *(pav 5)* toks email adresas jau nevalidus...

Dar viena įsivėlusi klaida – paduodamas tas pats adresas, neimama iš @ValueSource:

```
@ParameterizedTest
@ValueSource(strings = {"test@email.com", "test@EMAIL.COM", "test@123.info", "test@1.com"})
void testEmailDomainName_true() { assertTrue(EmailValidator.validateServer( email: "test@.com")); }
@ParameterizedTest
@ValueSource(strings = {"test@","test@#!email.com", "test@eeeeemmmmmaaaaaiiiiillllll.aaa", "test@12.com",
        "test@eeeeemmmmmaaaaaiiiiillllleeeeemmmmmaaaaaiiiiillllleeeeemmmmmaaaaaiiiiillllll.com"})
void testEmailDomainName_false() { assertFalse(EmailValidator.validateServer( email: "test@.com")); }
```

pav. 6 klaidingas @ParameterizedTest naudojimas.

Čia testai taip pat dviprasmiški. Pirmame metode test@1.com yra validus adresas, o antrame test@12.com jau nebevalidus. *(pav 6)*

3. Nekorektiški testai.

- TLD gali sudaryti tas pats alfabeto simbolis, tačiau tld ilgis negali būti trumpesnis nei 2 raidės, tad šis testas yra klaidingas, jis turėtų grąžinti tiesą. *(pav 7)*

```
@ParameterizedTest
@ValueSource(strings = {"test@email.aaaa"})
void testEmailTLD_false(String email) { assertFalse(EmailValidator.validateTLD(email)); }
```

pav. 7

- Domain name turi būti mažesnis nei 253, jį gali sudaryti raidės [a-z] arba skaičiai [0-9], šie simboliai gali kartotis ir būti skiriami . ar -. *Pav 8* pabraukti testai atitinka

reikalavimus ir yra teisingi.

```java
@ValueSource(strings = {"test@","test@#!email.com", "test@eeeeemmmmmaaaaaiiiiilllll.aaa", "test@12.com",
        "test@eeeeemmmmmaaaaaiiiiilllllleeeeemmmmmaaaaaiiiiilllllleeeeemmmmmaaaaaiiiiillllll.com"})
void testEmailDomainName_false(String email){
    assertFalse(EmailValidator.validateServer(email));
}
```

pav. 8