# Отчёт по выжившим тестам

*БПИ196 Татаринов Никита*

**Мутант 2**

В методе \_\_init\_\_(self):

self.\_\_blocked: bool = False

->

self.\_\_blocked: bool = None

Ошибка не была выявлена из-за того, что строка

assert not account.is\_blocked()

при значении None не выдавала ошибку.

Для исправления необходимо каждую такую строку заменить на

assert account.is\_blocked() is False

**Мутант 11**

В методе deposit(self, \_sum: int) -> bool:

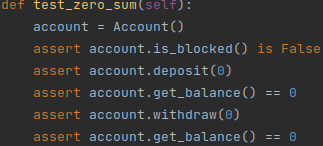
if \_sum < 0 or \_sum > self.\_\_bound:

->

if \_sum <= 0 or \_sum > self.\_\_bound:

Ошибка не была выявлена из-за того, что ни в одном тесте не проверялась попытка пользователя внести на счёт нулевую сумму.

Для исправления необходимо написать следующий тест (он также устраняет аналогичную ошибку у метода withdraw):



**Мутант 12**

В методе deposit(self, \_sum: int) -> bool:

if \_sum < 0 or \_sum > self.\_\_bound:

->

if \_sum < 1 or \_sum > self.\_\_bound:

Ошибка возникала по причине, аналогичной у мутанта 11, причём решение аналогичное.

**Мутант 13**

В методе deposit(self, \_sum: int) -> bool:

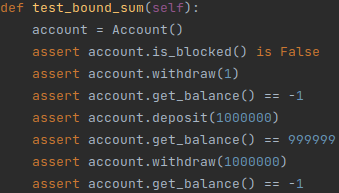
if \_sum < 0 or \_sum > self.\_\_bound:

->

if \_sum < 0 or \_sum >= self.\_\_bound:

Ошибка не была выявлена из-за того, что ни в одном тесте не проверялась попытка пользователя внести на счёт ровно граничную сумму денег.

Для исправления необходимо написать следующий тест (он также устраняет аналогичную ошибку у метода withdraw):



**Мутант 19**

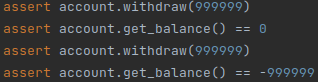
В методе deposit(self, \_sum: int) -> bool:

self.\_\_balance += \_sum:

->

self.\_\_balance = \_sum:

Ошибка не была выявлена из-за того, что ни в одном тесте не проверялись 2 успешные идущие подряд попытки внести на счёт ненулевую сумму денег. Важный момент: аналогичной ошибки для метода withdraw(self, \_sum: int) -> bool не выявилось. Связано это с тем, что в test8(self) проверяется эта ситуация.



Для исправления необходимо заменить строку



на строки



**Мутант 23**

В методе withdraw(self, \_sum: int) -> bool:

if \_sum < 0 or \_sum > self.\_\_bound:

->

if \_sum <= 0 or \_sum > self.\_\_bound:

Ошибка возникала по причине, аналогичной у мутанта 11, причём решение аналогичное.

**Мутант 24**

В методе withdraw(self, \_sum: int) -> bool:

if \_sum < 0 or \_sum > self.\_\_bound:

->

if \_sum < 1 or \_sum > self.\_\_bound:

Ошибка возникала по причине, аналогичной у мутанта 11, причём решение аналогичное.

**Мутант 25**

В методе deposit(self, \_sum: int) -> bool:

if \_sum < 0 or \_sum > self.\_\_bound:

->

if \_sum < 0 or \_sum >= self.\_\_bound:

Ошибка возникала по причине, аналогичной у мутанта 13, причём решение аналогичное.

**Мутант 37**

В методе unblock(self) -> bool:

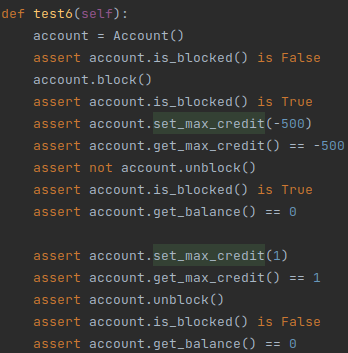
if self.\_\_balance < self.\_\_max\_credit:

->

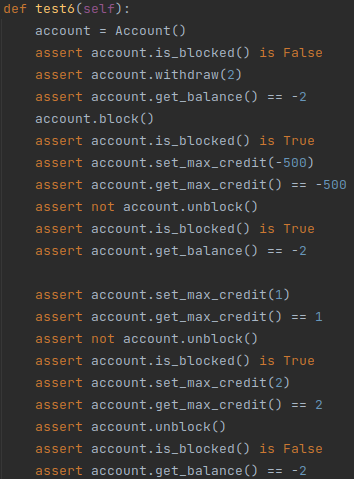
if self.\_\_balance <= self.\_\_max\_credit:

Ошибка не была выявлена из-за того, что ни в одном тесте не проверялось точное равенство между балансом и максимальным значением кредита при попытке разблокировать счёт.

Для исправления необходимо заменить код test6(self) с



на



**Мутант 40**

В методе unblock(self) -> bool:

self.\_\_blocked = False

->

self.\_\_blocked = None

Ошибка возникала по причине, аналогичной у мутанта 11, причём решение аналогичное.