# Отчёт по методам

*БПИ196 Татаринов Никита*

**Метод \_\_init\_\_(self)**

Возьмём 1й мутант:

self.\_\_blocked: bool = False

->

self.\_\_blocked: bool = True

Тесты не проходят из-за того, что аккаунт оказывается изначально заблокированным и не работают денежные операции.

**Метод deposit(self, \_sum: int) -> bool**

Возьмём 16й мутант:

if self.\_\_balance >= self.\_\_bound - \_sum:

->

if self.\_\_balance > self.\_\_bound - \_sum:

Тесты не проходят из-за того, что верхняя граница self.\_\_bound становится доступной для пользователя, чего не должно происходить.

**Метод withdraw(self, \_sum: int) -> bool**

Возьмём 27й мутант:

return False

->

return True

Тесты не проходят из-за того, что пользователь информируется о том, что транзакция проведена успешна, несмотря на обратное.

**Метод get\_balance(self) -> int**

В данном методе не присутствует мутантов.

**Метод get\_max\_credit(self) -> int**

Возьмём 34й мутант:

return -self.\_\_max\_credit

->

return +self.\_\_max\_credit

Тесты не проходят из-за того, что пользователю возвращается отрицательное максимального значение кредита, что с точки зрения логики пользователя неверно.

**Метод is\_blocked(self) -> bool**

В данном методе не присутствует мутантов.

**Метод block(self) -> None**

Возьмём 35й мутант:

self.\_\_blocked = True

->

self.\_\_blocked = False

Тесты не проходят из-за того, что при попытке пользователя заблокировать аккаунт он, наоборот, разблокируется, что диаметрально противоположно логике блокирования.

**Метод unblock(self) -> bool**

Возьмём 41й мутант:

return True

->

return False

Тесты не проходят из-за того, что аккаунт становится заблокированным, а пользователю возвращается информация о том, что он остался незаблокированным.

**Метод set\_max\_credit(self, mc: int) -> bool**

Возьмём 44й мутант:

if abs(mc) > self.\_\_bound:

->

if abs(mc) >= self.\_\_bound:

Тесты не проходят из-за того, пользователю предоставляется возможность установить максимальное значение кредита в величину границе, что по логике должно быть недоступно.

**Эквивалентный мутант**

В методе \_\_init\_\_(self):

self.\_\_max\_credit: int = -1000

->

self.\_\_max\_credit: int = -100

С точки зрения логики ничего не меняется: у пользователя всё так же есть максимальное значение кредита, то есть меняются только суммы, с которыми может оперировать пользователь, но сами операции не меняются.