Кирилл Дудников. БПИ206.

**Структура домашней работы:**

* в папке pit-reports-before лежит репорт до исправления тестов
* в папке pit-reports-after лежит репорт после исправления тестов
* в файле VendingMachineTest.java лежат доработанные тесты (добавлены методы giveProduct1CorrectMaxProducts() и giveProduct2CorrectMaxProducts())
* в этом отчете лежат объяснения для непойманных мутантов и “по 1 примеру мутанта для каждого метода, а также 1 на класс пример эквивалентного мутанта”

**Выжившие мутанты:**

Среди них красным отмечаются выжившие неидентичные мутанты (т.е. реально непроверенные случаи).

М79.1. Метод getCurrentSum()

1. return coins1\*coinval1+coins2\*coinval2;
2. return coins1/coinval1+coins2\*coinval2;
3. Не убитый
4. Этот мутант идентичен, так как coinval1 всегда = 1 и черным ящиком это никак не изменить.

М179.2. Метод returnMoney()

1. else if(balance > coins1\*coinval1 + coins2\*coinval2)
2. else if(balance > coins1/coinval1 + coins2\*coinval2)
3. Не убитый
4. Идентичный мутант, т.к. coinval1 всегда = 1.

М181.1. Метод returnMoney()

1. return Response.TOO\_BIG\_CHANGE;
2. return null;
3. Не убитый (NO\_COVERAGE)
4. Идентичный мутант. Объяснял в предыдущей работе - в этот участок кода невозможно зайти черным ящиком.

М183.1. Метод returnMoney()

1. else if(balance > coins2\*coinval2)
2. else if(balance >= coins2\*coinval2)
3. Не убитый
4. Идентичный мутант. Если balance == coins2\*coinval2, то и в ветке “else if(balance > coins2\*coinval2)” и в ветке “else if(balance%coinval2 == 0)” будет корректно обрабатываться результат. Всегда будет coins1 -= 0, coins2 = 0, balance = 0.

М202.1. Метод returnMoney()

1. return Response.UNSUITABLE\_CHANGE;
2. return null;
3. Не убитый (NO\_COVERAGE)
4. Идентичный мутант. Объяснял в предыдущей работе - в этот участок кода невозможно зайти черным ящиком.

М219.2. Метод giveProduct1(int number)

1. if(number <= 0 || number > max1) return Response.INVALID\_PARAM;
2. if(number <= 0 || number >= max1) return Response.INVALID\_PARAM;
3. Не убитый
4. Недостаточность моих тестов. Исправлено.

М220.1. Метод giveProduct1(int number)

1. if(number > num1) return Response.INSUFFICIENT\_PRODUCT;
2. if(number >= num1) return Response.INSUFFICIENT\_PRODUCT;
3. Не убитый
4. Недостаточность моих тестов. Исправлено.

М225.1. Метод giveProduct1(int number)

1. else if(res > coins1\*coinval1 + coins2\*coinval2)
2. else if(res >= coins1\*coinval1 + coins2\*coinval2)
3. Не убитый
4. Идентичный мутант.   
   Знаем, что:  
   res = balance - number\*price1.  
   Логично, что: balance <= coins1\*coinval1 + coins2\*coinval2  
   Тогда: balance - number\*price1 <= coins1\*coinval1 + coins2\*coinval2  
   Но ни price1, ни number не может быть = 0 (по ТЗ, когда устанавливается цена на продукт1, она != 0, а при покупке можно нельзя купить 0 продуктов1).  
   Значит, нестрогое неравенство превращается в строгое.  
   balance - number\*price1 < coins1\*coinval1 + coins2\*coinval2  
   То же самое, что: res < coins1\*coinval1 + coins2\*coinval2  
   А значит, изменение “res > coins1\*coinval1 + coins2\*coinval2”   
   на “res >= coins1\*coinval1 + coins2\*coinval2” ничего не изменит, в эту ветку все равно нельзя будет зайти.

М225.2. Метод giveProduct1(int number)

1. else if(res > coins1\*coinval1 + coins2\*coinval2)
2. else if(res > coins1/coinval1 + coins2\*coinval2)
3. Не убитый
4. Идентичный мутант, т.к. coinval1 всегда = 1.

М227.1. Метод giveProduct1(int number)

1. return Response.TOO\_BIG\_CHANGE;
2. return null;
3. Не убитый (NO\_COVERAGE)
4. Идентичный мутант. В этот участок кода невозможно зайти черным ящиком.

М229.1. Метод giveProduct1(int number)

1. else if(res > coins2\*coinval2)
2. else if(res >= coins2\*coinval2)
3. Не убитый
4. Идентичный мутант. Если res == coins2\*coinval2, то и в ветке “else if(res > coins2\*coinval2)” и в ветке “else if(res%coinval2 == 0)” будет корректно обрабатываться результат. Всегда будет coins1 -= 0, coins2 = 0, balance = 0, num1 -= number.

М268.2. Метод giveProduct2(int number)

1. if(number <= 0 || number > max2) return Response.INVALID\_PARAM;
2. if(number <= 0 || number >= max2) return Response.INVALID\_PARAM;
3. Не убитый
4. Недостаточность моих тестов. Исправлено.

М269.1. Метод giveProduct2(int number)

1. if(number > num2) return Response.INSUFFICIENT\_PRODUCT;
2. if(number >= num2) return Response.INSUFFICIENT\_PRODUCT;
3. Не убитый
4. Недостаточность моих тестов. Исправлено.

М274.1. Метод giveProduct2(int number)

1. else if(res > coins1\*coinval1 + coins2\*coinval2)
2. else if(res >= coins1\*coinval1 + coins2\*coinval2)
3. Не убитый
4. Идентичный мутант. Доказывается аналогично с М225.1.

М274.2. Метод giveProduct2(int number)

1. else if(res > coins1\*coinval1 + coins2\*coinval2)
2. else if(res > coins1/coinval1 + coins2\*coinval2)
3. Не убитый
4. Идентичный мутант, т.к. coinval1 всегда = 1.

М276.1. Метод giveProduct2(int number)

1. return Response.TOO\_BIG\_CHANGE;
2. return null;
3. Не убитый (NO\_COVERAGE)
4. Идентичный мутант. В этот участок кода невозможно зайти черным ящиком.

М278.1. Метод giveProduct2(int number)

1. else if(res > coins2\*coinval2)
2. else if(res >= coins2\*coinval2)
3. Не убитый
4. Идентичный мутант. Доказывается аналогично с М229.1.

**“по 1 примеру мутанта для каждого метода”**

Всего методов 19 штук.

Pitest прекращает тестирование мутанта сразу после того, как какой-либо тест убивает мутанта. Так что в отчете я пишу только первый метод, убивший мутанта.

М56. Метод getNumberOfProduct1()

1. return num1;
2. return 0;
3. Убитый
4. giveProduct1InAdministratorMode()

М61. Метод getNumberOfProduct2()

1. return num2;
2. return 0;
3. Убитый
4. giveProduct2InAdministratorMode()

М66. Метод getCurrentBalance()

1. return balance;
2. return 0;
3. Убитый
4. putCoin2PutMaxOk()

М71. Метод getCurrentMode()

1. return mode;
2. return null;
3. Убитый
4. enterAdminModeWrongCode()

М76. Метод getCurrentSum()

1. if(mode == Mode.OPERATION)
2. if(mode != Mode.OPERATION)
3. Убитый
4. getCurrentSumInOperationMode()

М84. Метод getCoins1()

1. if(mode == Mode.OPERATION)
2. if(mode != Mode.OPERATION)
3. Убитый
4. getCoins1InOperationMode()

М92. Метод getCoins2()

1. if(mode == Mode.OPERATION)
2. if(mode != Mode.OPERATION)
3. Убитый
4. getCoins2InOperationMode()

М100. Метод getPrice1()

1. return price1;
2. return 0;
3. Убитый
4. setPricesInOperationMode()

М105. Метод getPrice2()

1. return price2;
2. return 0;
3. Убитый
4. setPricesInOperationMode()

М110. Метод fillProducts()

1. if(mode == Mode.OPERATION) return Response.ILLEGAL\_OPERATION;
2. if(mode != Mode.OPERATION) return Response.ILLEGAL\_OPERATION;
3. Убитый
4. giveProduct1InAdministratorMode()

М119. Метод fillCoins(int c1, int c2)

1. if(c1 <= 0 || c1 > maxc1) return Response.INVALID\_PARAM;
2. if(c1 < 0 || c1 > maxc1) return Response.INVALID\_PARAM;
3. Убитый
4. fillCoinsWrongParams()

М128. Метод enterAdminMode(long code)

1. if(balance != 0) return Response.CANNOT\_PERFORM;
2. if(balance != 0) return null;
3. Убитый
4. enterAdminModeNonZeroBalanceCoin1()

М(номера нет - Pitest не сделал мутацию). Метод exitAdminMode()

1. mode = Mode.OPERATION;
2. mode = Mode.ADMINISTERING;
3. Убитый
4. fillProductsInOperationMode()

М143. Метод setPrices(int p1, int p2)

1. if(p1 <= 0 || p2 <= 0) return Response.INVALID\_PARAM;
2. if(p1 <= 0 || p2 < 0) return Response.INVALID\_PARAM;
3. Убитый
4. setPricesWrongParams()

М154. Метод putCoin1()

1. balance += coinval1;
2. balance -= coinval1;
3. Убитый
4. putCoin1PutMaxOk()

М166. Метод putCoin2()

1. coins2++;
2. coins2--;
3. Убитый
4. returnMoneyZeroCoinsInitially()

М186. Метод returnMoney()

1. coins1 -= (balance-coins2\*coinval2);
2. coins1 -= (balance-coins2/coinval2);
3. Убитый
4. returnMoneyCondition1()

М224. Метод giveProduct1(int number)

1. if(res < 0) return Response.INSUFFICIENT\_MONEY;
2. if(res <= 0) return Response.INSUFFICIENT\_MONEY;
3. Убитый
4. putCoin2PutOneOk()

М292. Метод giveProduct2(int number)

1. num2 -= number;
2. num2 += number;
3. Убитый
4. giveProduct2CorrectCondition2()

**“1 на класс пример эквивалентного мутанта”**

М274.1. Метод giveProduct2(int number)

1. else if(res > coins1\*coinval1 + coins2\*coinval2)
2. else if(res >= coins1\*coinval1 + coins2\*coinval2)
3. Не убитый