

Ошибка 1

1. ФИО обнаружившего ошибку – Куликова Татьяна Дмитриевна
2. Дата и время обнаружения - 01.10.22, 16:00
3. Метод `getCoins2` не всегда возвращает 0 в режиме `Operation`.
4. Ошибка нарушает требование `g`.
5. Если в момент вызова метода `getCoins2` система находится в режиме `Operation`, а значение переменной `coins1` не равно 0, возвращаемое значение не удовлетворяет требованию.
6. **Тест:**

```
VendingMachine machine = new VendingMachine();  
machine.enterAdminMode(117345294655382L);  
machine.fillCoins(1, 1);  
machine.exitAdminMode();  
assertEquals(0, machine.getCoins2());
```
7. Первоначальная строка: `return coins1;`
Исправленная строка: `return 0;`

Ошибка 2

1. ФИО обнаружившего ошибку – Куликова Татьяна Дмитриевна
2. Дата обнаружения ошибки – 01.10.22, 16:30
3. Метод `fillProducts` устанавливает значение количества первого продукта равным максимально возможному количеству второго продукта (а не первого).
4. Ошибка нарушает требование `j`.
5. Если установленные максимальные значения количества первого и второго продукта не равны, метод придает переменной `num1` неправильное значение, равное `max2`.
6. **Тест:**

```
VendingMachine machine = new VendingMachine();  
machine.enterAdminMode(117345294655382L);  
assertEquals(VendingMachine.Response.OK,  
machine.fillProducts());  
assertEquals(30, machine.getNumberOfProduct1());  
assertEquals(40, machine.getNumberOfProduct2());  
// при max1 = 30 и max2 = 40.
```
7. Первоначальная строка: `num1 = max2;`
Исправленная строка: `num1 = max1;`

Ошибка 3

1. ФИО обнаружившего ошибку – Куликова Татьяна Дмитриевна
2. Дата обнаружения ошибки – 01.10.22, 17:00
3. При значении аргумента `C1 > maxC1` и некоторых значениях аргумента `C2` метод `fillCoins` возвращает `Response OK` вместо `INVALID_PARAM`.
4. Ошибка нарушает требование `k`.
5. Если `C1 > maxC1`, а `C2 <= maxC1` и `C2 <= maxC2`, метод возвращает `Response OK` вместо `INVALID_PARAM`.
6. **Тест:**

```
VendingMachine machine = new VendingMachine();
```

```
machine.enterAdminMode(117345294655382L);  
assertEquals(VendingMachine.Response.INVALID_PARAM  
machine.fillCoins(52, 50));
```

7. Первоначальная строка: `if (c1 <= 0 || c2 > maxc1) return Response.INVALID_PARAM;`
Исправленная строка: `if (c1 <= 0 || c1 > maxc1) return Response.INVALID_PARAM;`

Ошибка 4

1. ФИО обнаружившего ошибку – Куликова Татьяна Дмитриевна
2. Дата обнаружения ошибки – 01.10.22, 17:30
3. При значении баланса, не равном нулю, метод `enterAdminMode` возвращает `Response UNSUITABLE_CHANGE` вместо `CANNOT_PERFORM`.
4. Ошибка нарушает требование 1.
5. Если `balance != 0`, метод `enterAdminMode` возвращает `Response UNSUITABLE_CHANGE` вместо `CANNOT_PERFORM`.
6. **Тест:**

```
VendingMachine machine = new VendingMachine();  
machine.putCoin1();  
assertEquals(VendingMachine.Response.CANNOT_PERFORM,  
machine.enterAdminMode(117345294655382L));
```
7. Первоначальная строка: `if(balance != 0) return Response.UNSUITABLE_CHANGE;`
Исправленная строка: `if(balance != 0) return Response.CANNOT_PERFORM;`

Ошибка 5

1. ФИО обнаружившего ошибку – Куликова Татьяна Дмитриевна
2. Дата обнаружения ошибки – 01.10.22, 18:00
3. При значении цены на второй продукт `p2 <= 0` и некоторых значениях `p1`, метод `setPrices` возвращает `Response OK` вместо `INVALID_PARAM`.
4. Ошибка нарушает требование п.
5. Если `p2 <= 0` и `p1 > 0`, метод `setPrices` возвращает `Response OK` вместо `INVALID_PARAM`, так как проверяется только параметр `p1`.
6. **Тест:**

```
VendingMachine machine = new VendingMachine();  
machine.enterAdminMode(117345294655382L);  
assertEquals(VendingMachine.Response.INVALID_PARAM,machine.set  
Prices(16, -10));
```
7. Первоначальная строка: `if(p1 <= 0 || p1 <= 0) return Response.INVALID_PARAM;`
Исправленная строка: `if(p1 <= 0 || p2 <= 0) return Response.INVALID_PARAM;`

Ошибка 6

1. ФИО обнаружившего ошибку – Куликова Татьяна Дмитриевна
2. Дата обнаружения ошибки – 01.10.22, 18:30
3. Метод `putCoin1` пополняет баланс пользователя монетой второго вида (вместо монеты первого вида)
4. Ошибка нарушает требование о.

5. Если значения `coinval1` и `coinval2` не равны, баланс пользователя после выполнения метода принимает неверное значение. Также если количества монет 1 и 2 вида не равны, то значения `coins1` и `coins2` после выполнения метода также примут неверные значения, так как значение `coins2` увеличивается, а `coins1` не изменяется.

6. **Тест:**

```
VendingMachine machine = new VendingMachine();  
machine.enterAdminMode(117345294655382L);  
machine.fillCoins(1, 3);  
machine.exitAdminMode();  
assertEquals(VendingMachine.Response.OK, machine.putCoin1());  
assertEquals(1, machine.getCurrentBalance());
```

7. **Фрагмент кода до исправления:**

```
If (coins2 == maxc2) return Response.CANNOT_PERFORM;  
    balance += coinval2;  
    coins2++;
```

Фрагмент кода после исправления:

```
If (coins1 == maxc1) return Response.CANNOT_PERFORM;  
    balance += coinval1;  
    coins1++;
```

Ошибка 7

1. ФИО обнаружившего ошибку – Куликова Татьяна Дмитриевна
2. Дата обнаружения ошибки – 01.10.22, 19:00
3. Метод `putCoin2` пополняет баланс пользователя монетой первого вида (вместо монеты второго вида)
4. Ошибка нарушает требование р.
5. Если значения `coinval1` и `coinval2` не равны, баланс пользователя после выполнения метода принимает неверное значение. Также если количества монет 1 и 2 вида не равны, то значения `coins1` и `coins2` после выполнения метода также примут неверные значения, так как значение `coins1` увеличивается, а `coins2` не изменяется.

6. **Тест:**

```
VendingMachine machine = new VendingMachine();  
machine.enterAdminMode(117345294655382L);  
machine.fillCoins(3, 1);  
machine.exitAdminMode();  
assertEquals(VendingMachine.Response.OK, machine.putCoin1());  
assertEquals(2, machine.getCurrentBalance());
```

7. **Фрагмент кода до исправления:**

```
If (coins1 == maxc1) return Response.CANNOT_PERFORM;  
    balance += coinval1;  
    coins1++;
```

Фрагмент кода после исправления:

```
If (coins2 == maxc2) return Response.CANNOT_PERFORM;  
    balance += coinval2;  
    coins2++;
```

Ошибка 8

1. ФИО обнаружившего ошибку – Куликова Татьяна Дмитриевна

2. Дата обнаружения ошибки – 01.10.22, 19:30
3. Метод giveProduct2 при невозможности выдачи сдачи имеющимися монетами возвращает Response INSUFFICIENT_MONEY (Вместо TOO_BIG_CHANGE).
4. Ошибка нарушает требование s.
5. Если значения balance, price2, number, coinval1, coinval2, coins1, coins2 такие, что $balance - number * price2 > coinval1 * coins1 + coinval2 * coins2$, метод возвращает Response UNSUFFICIENT_MONEY (вместо TOO_BIG_CHANGE).
6. **Тест:**

```
VendingMachine machine = new VendingMachine();
machine.enterAdminMode(117345294655382L);
machine.fillCoins(1, 1);
machine.setPrices(1, 2);
machine.fillProducts();
machine.exitAdminMode();
VendingMachine.coinval2 = 5;
VendingMachine.coinval1 = 5;
for (int i = 0; i < 3; ++i) {
    machine.putCoin2();
}
VendingMachine.coinval2 = 1;
VendingMachine.coinval1 = 1;
assertEquals(VendingMachine.Response.TOO_BIG_CHANGE,
machine.giveProduct2(1));
```

7. **Фрагмент кода до исправления:**

```
else if(res > coins1*coinval1 + coins2*coinval2)
{
    return Response.UNSUFFICIENT_MONEY;
}
```

Фрагмент кода после исправления:

```
else if(res > coins1*coinval1 + coins2*coinval2)
{
    return Response.TOO_BIG_CHANGE;
}
```

Ошибка 9

1. ФИО обнаружившего ошибку – Куликова Татьяна Дмитриевна
2. Дата обнаружения ошибки – 01.10.22, 20:00
3. Метод returnMoney в некоторых ситуациях работает неправильно – после возврата баланса придает переменным coins1 и coins2 неверные значения.
4. Ошибка нарушает требование q.
5. Если значение $balance \leq coins1 * coinval1 + coins2 * coinval2$, $balance \leq coins2 * coinval2$, $balance \% coinval2 \neq 0$ и $coins1 > 0$, то монеты выдаются не в соответствии с требованиями: в первую очередь выдаются монеты 1 вида, а оставшаяся часть выдается монетами второго вида, причем монеты каждого вида выдаются в неправильном количестве.
6. **Тест:**

```
VendingMachine machine = new VendingMachine();
machine.enterAdminMode(117345294655382L);
machine.fillCoins(1, 5);
machine.exitAdminMode();
```

```

for (int i = 0; i < 4; ++i) {
    machine.putCoin2();
}
machine.putCoin1();
assertEquals(VendingMachine.Response.OK,
machine.returnMoney());
assertEquals(0, machine.getCurrentBalance());
machine.enterAdminMode(117345294655382L);
assertEquals(5, machine.getCoins2());
assertEquals(1, machine.getCoins1());

```

7. Фрагмент кода до исправления:

```

coins1 -= (balance/coinval2);
coins2--;

```

Фрагмент кода после исправления:

```

coins2 -= (balance/coinval2);
balance = balance % coinval2;
coins1 -= (balance / coinval1);

```

Ошибка 10

1. ФИО обнаружившего ошибку – Куликова Татьяна Дмитриевна
2. Дата обнаружения ошибки – 01.10.22, 20:30
3. Метод `giveProduct2` в некоторых ситуациях работает неправильно – после возврата баланса придает переменным `coins1` и `coins2` неверные значения.
4. Ошибка нарушает требование `s`.
5. Если значение $res \leq coins1 * coinval1 + coins2 * coinval2$, $res \leq coins2 * coinval2$, $res \% coinval2 \neq 0$ и $coins1 > 0$, где $res = balance - number * price2$, то монеты выдаются не в соответствии с требованиями: в первую очередь выдаются монеты 1 вида, а оставшаяся часть выдается монетами второго вида, причем монеты каждого вида выдаются в неправильном количестве.

6. Тест:

```

VendingMachine machine = new VendingMachine();
machine.enterAdminMode(117345294655382L);
machine.fillCoins(1, 5);
machine.fillProducts();
machine.setPrices(2, 1);
machine.exitAdminMode();
for (int i = 0; i < 3; ++i) {
    machine.putCoin2();
}
machine.putCoin1();
assertEquals(VendingMachine.Response.OK,
machine.giveProduct2(2));
assertEquals(0, machine.getCurrentBalance());
machine.enterAdminMode(117345294655382L);
assertEquals(6, machine.getCoins2());
assertEquals(1, machine.getCoins1());

```

7. Фрагмент кода до исправления:

```

coins1 -= (res/coinval2);
coins2--;

```

Фрагмент кода после исправления:

```
coins2 -= (res/coinval2);  
res = res % coinval2;  
coins1 -= (res / coinval1);
```

Ошибка 11

1. ФИО обнаружившего ошибку – Куликова Татьяна Дмитриевна
2. Дата обнаружения ошибки – 01.10.22, 21:00
3. В методе fillProducts нет проверки на текущий режим.
4. Ошибка нарушает требование j.
5. Если установленный режим – это режим Operation, то метод в соответствии с требованием j должен возвращать Response ILLEGAL_OPERATION, однако он возвращает Response OK.

6. **Тест:**

```
VendingMachine machine = new VendingMachine();  
assertEquals(VendingMachine.Response.ILLEGAL_OPERATION,  
machine.fillProducts());
```

7. Первоначальная фрагмент кода:

```
num1 = max1;  
num2 = max2;  
return Response.OK;
```

Исправленная фрагмент кода:

```
if(mode == Mode.OPERATION) return Response.ILLEGAL_OPERATION;  
num1 = max1;  
num2 = max2;  
return Response.OK;
```