**Список обнаруженных ошибок и исправлений**

**Выполнил Игумнов Никита БПИ191**

1. Код до исправления:

public boolean setMaxCredit(int mc) {  
 // *TODO: FIX: В незаблокированном состоянии запрещено изменять значение кредитного максимума* if (mc < -bound || mc > bound)  
 return false;  
 else  
 maxCredit = -mc;  
  
 return true;  
}

2. При незаблокированном состоянии и входным аргументом меньшем или равным по модулю граничного значения.

3. Полученное значение: **true**. Требуемое значение: **false**.

4. Код после исправлений:

public boolean setMaxCredit(int maxCredit) {  
 if (!isBlocked()) {  
 return false;  
 }  
  
 if (maxCredit < -MAX\_BOUND || maxCredit > MAX\_BOUND) {  
 return false;  
 }  
  
 this.maxCredit = maxCredit;  
 return true;  
}

1. Код до исправления:

public boolean unblock() {  
 // *TODO: FIX: maxCredit with minus.* if (balance < maxCredit)  
 return false;  
 else  
 blocked = false;  
  
 return true;  
}

2. Если maxCredit ≤ balance < -maxCredit

3. Полученное значение: **true**. Требуемое значение: **false**.

4. Код после исправлений:

public boolean unblock() {  
 if (balance < -maxCredit) {  
 return false;  
 }  
  
 blocked = false;  
 return true;  
}

1. Код до исправления:

public boolean withdraw(int sum) {  
 if (blocked) {  
 return false;  
 } else if (sum < 0 || sum > bound) {  
 return false;  
 } else if (balance <= maxCredit + sum) {  
 return false;  
 } else {  
 balance -= sum;  
 return true;  
 }  
}

2. Если 0 ≤ sum ≤ 1000000 и maxCredit < balance - sum < -maxCredit

3. Полученное значение: **false** без изменения баланса. Требуемое значение: **true** с изменённым балансом.

4. Код после исправлений:

public boolean withdraw(int sum) {  
 if (isBlocked()) {  
 return false;  
 } else if (sum < 0 || sum > MAX\_BOUND) {  
 return false;  
 } else if (balance - sum < -maxCredit) {  
 return false;  
 } else {  
 balance -= sum;  
 return true;  
 }  
}