

```
private void button1_Click(object sender, EventArgs e)
{
    Buffer += "1";
    if (operationSemaphore == OperationType.PIN)
        MainScreen.displayMessage("*");
    else if (operationSemaphore == OperationType.DRAW)
        MainScreen.displayMessage("1");
}
```

```
private void button2_Click(object sender, EventArgs e)
{
    Buffer += "2";
    if (operationSemaphore == OperationType.PIN)
        MainScreen.displayMessage("*");
    else if (operationSemaphore == OperationType.DRAW)
        MainScreen.displayMessage("2");
}
```

1. } - и таких аж 10 функций, должно быть копияст... Неудачное решение, можно было бы реализовать одно событие нажатия для нескольких кнопок сразу.

```
public ATM_Mainform()
{
    InitializeComponent();
    toScreen = richTextBox1;
    toListBox = listBox1;
    ATMObject = new ATM();
    __thread = new Thread(new ThreadStart( StartATM ));
    __thread.IsBackground = true;
    __thread.Start();
}
```

2. } - можно ж было сразу переменные обозвать как надо, ну зачем еще лишние переменные заводить?

```
//asdasdds
```

3. `delegate void textDelegate(String text);` - очень понятный комментарий

4. Зачем-то объект класса банкомат работает в отдельном потоке, это как из пушки по воробьям.

```
private void getListBoxIndex()
{
    selectedIndex = toListBox.SelectedIndex;
}
```

5. } - геттер возвращает void, и там еще такие функции есть.

```

private void CheckCanceled()
{
    keySemaphore = KeyType.NOTHING;
    do
    {
        Thread.Sleep(200);
    } while (keySemaphore != KeyType.CANCEL);
    keySemaphore = KeyType.NOTHING;
    userAuthenticated = false;
    currentPIN = 0;
    __thread.Abort();
    screen.displayMessage("\nВыход из системы...");
}

```

6. - опять потоки, в классе банкомат (который и так в отдельном потоке) есть поток, который проверяет, не нажал ли юзер CANCEL.

```

if (keySemaphore == KeyType.BACK)
    return -2;

```

7. - магические числа.

```

private int displayMenuOfAmounts()
{
    int userChoice = 0;
    Screen screen = getScreen();
    List<int> amounts = new List<int>();
    amounts.Add( 0 );
    amounts.Add( 100 );
    amounts.Add( 500 );
    amounts.Add( 1000 );
    amounts.Add( 5000 );
}

```

8. - каждый раз, когда вызывается эта функция, создается новый список amounts, и затем заполняется. Лучше бы сделать его статическим, да и вообще не список использовать, а массив.

```

public void credit(int userPIN, double amount)
{
    Account userAccountPtr = getAccount(userPIN);
    userAccountPtr.credit(amount);
}
public void debit(int userPIN, double amount)
{
    Account userAccountPtr = getAccount(userPIN);
    userAccountPtr.debit(amount);
}

private List<Account> accounts = new List<Account>();

private Account getAccount(int userPIN)
{
    for (int i = 0; i < accounts.Count; i++)
    {
        if (accounts[i].getUserPIN() == userPIN)
            return accounts[i];
    }
    return null;
}

public void getListOfAccounts( System.Windows.Forms.ListBox listbox )
{
    for (int i = 0; i < 30; i++)
        listbox.Items.Add(accounts[i].toString());
}

```

9.

описание переменной посреди описаний методов.

```

public class DepositSlot
{
    public bool isEnvelopeReceived()
    {
        return true;
    }
}

```

10. Ну и напоследок: