

Spisak korektivnih akcija

Za svakog člana tima su navedene korektivne akcije za njegov dio koda.

Program.cs

1. Ispravljena greška u liniji 42 pri konvertovanju bajtova u string u funkciji HashFunction.

```
public static string HashFunction(string pin)
{
    using (SHA256 sha256 = SHA256.Create())
    {
        byte[] pinBytes = Encoding.UTF8.GetBytes(pin);
        byte[] hashBytes = sha256.ComputeHash(pinBytes);
        string pinHash = BitConverter.ToString(hashBytes).ToLower();
        return pinHash;
    }
}

public static string HashFunction(string pin)
{
    using (SHA256 sha256 = SHA256.Create())
    {
        byte[] pinBytes = Encoding.UTF8.GetBytes(pin);
        byte[] hashBytes = sha256.ComputeHash(pinBytes);
        string pinHash = BitConverter.ToString(hashBytes).Replace("-", "").ToLower();
        return pinHash;
    }
}
```

2. Ispravljena greška u liniji 95 pri resetovanju vrijednosti u funkciji CheckUserCredentials, varijabla isCorrectLogin korektno resetovana na vrijednost true.

```
foreach (User account in userList)
{
    selectedUser = account;
    string provjera = HashFunction(selectedUser.CardPin);
    inputAccount.CardPin = provjera;
    if (inputAccount.CardNumber.Equals(selectedUser.CardNumber))
    {
        if (inputAccount.CardPin.Equals(provjera))
        {
            selectedUser = account;
            break;
        }
    }
}
```

```

foreach (User account in userList)
{
    selectedUser = account;
    string provjera = HashFunction(selectedUser.CardPin);
    inputAccount.CardPin = provjera;
    if (inputAccount.CardNumber.Equals(selectedUser.CardNumber))
    {
        if (inputAccount.CardPin.Equals(provjera))
        {
            selectedUser = account;
            isCorrectLogin = true;
            break;
        }
    }
}

```

3. Ispravljena greška u liniji 184 u funkciji ProcessMenuChoice tj. dodana provjera na izuzetak koji baca funkcija ProcessInternalTransfer.

```

case 4:
    if (screen == null)
    {
        screen = new ATMScreen();
    }
    var internalTransfer = screen.CreateInternalTransferTransaction();
    ProcessInternalTransfer(internalTransfer);
    break;

```

```

case 4:
    if (screen == null)
    {
        screen = new ATMScreen();
    }
    var internalTransfer = screen.CreateInternalTransferTransaction();
    try
    {
        ProcessInternalTransfer(internalTransfer);
    }
    catch (Exception ex)
    {
        Utility.PrintMessage(ex.Message, false);
    }
    break;

```

4. Ispravljena greška u liniji 296 u funkciji MakeWithdrawal, poredjenje korektno prepravljeno tako da varijabla transactionAmount ne samo da mora biti nenegativna već mora biti strogo pozitivna.

```
if (transactionAmount < 0)
```

```
if (transactionAmount <= 0)
```

5. Ispravljena greška u liniji 413 u funkciji ViewTransaction tako da case 1 služi za sortiranje po iznosu a case 2 služi za sortiranje po tipu.



```
407 case "1":
408     Console.WriteLine("\nSorting by type...");
409     Utility.PrintDotAnimation(false);
410     table = sortTableByType();
411     Utility.PressEnterToContinue();
412     break;
413 case "2":
414     Console.WriteLine("\nSorting by amount...");
415     Utility.PrintDotAnimation(false);
416     table = sortTableByAmount();
417     Utility.PressEnterToContinue();
418     break;
419 default:
420     break;
```

```
407 case "1":
408     Console.WriteLine("\nSorting by amount...");
409     Utility.PrintDotAnimation(false);
410     table = sortTableByAmount();
411     Utility.PressEnterToContinue();
412     break;
413 case "2":
414     Console.WriteLine("\nSorting by type...");
415     Utility.PrintDotAnimation(false);
416     table = sortTableByType();
417     Utility.PressEnterToContinue();
418     break;
419 default:
420     break;
```

ATMScreen.cs

1. Ispravljena greška u liniji 26 u funkciji UserLogin, pri konvertovanju unosa broja kartice sa konzole, sa int prepravljen u long.

```
22 public static User UserLogin()  
23 {  
24     User tempUser = new User();  
25  
26     tempUser.CardNumber = Utility.Convert<int>("Enter your card number.");  
27     tempUser.CardPin = Convert.ToString(Utility.GetPINInput("Enter your card PIN"));  
28     Program.HashFunction(tempUser.CardPin);  
29     return tempUser;  
30 }
```

```
22 public static User UserLogin()  
23 {  
24     User tempUser = new User();  
25  
26     tempUser.CardNumber = Utility.Convert<long>("Enter your card number.");  
27     tempUser.CardPin = Convert.ToString(Utility.GetPINInput("Enter your card PIN"));  
28     Program.HashFunction(tempUser.CardPin);  
29     return tempUser;  
30 }
```

2. Ispravljena greška u liniji 40 u funkciji WelcomeCustomer, dodana metoda za ispis imena korisnika.

```
38 public static void WelcomeCustomer(string fullName)  
39 {  
40  
41     Utility.PressEnterToContinue();  
42 }
```

```
38 public static void WelcomeCustomer(string fullName)  
39 {  
40     Console.WriteLine($"Welcome {fullName}");  
41     Utility.PressEnterToContinue();  
42 }
```

3. Ispravljena greška u liniji 119 u funkciji SelectWithdrawalAmount, korektno dodan default slučaj ako se desi da korisnik pogrešno odabere iznos.

```
115 case 8:  
116     return 10000;  
117 case 0:  
118     return 0;  
119 default:  
120     return -1;  
121 }  
122 }
```

```
115 case 8:  
116     return 10000;  
117 case 0:  
118     return 0;  
119 default:  
120     Utility.PrintMessage("Invalid input. Try again.", false);  
121     return -1;  
122 }  
123 }
```

Utility.cs

1. Ispravljena greška u liniji 76 u funkciji PrintMessage, dodana provjera ako je pritisnut enter poziva se funkcija PressEnterToContinue.

```
16 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
62 public static void PrintMessage(string msg, bool success = true, bool pressEnter = true)
63 {
64     if (success)
65     {
66         Console.ForegroundColor = ConsoleColor.Yellow;
67     }
68     else
69     {
70         Console.ForegroundColor = ConsoleColor.Red;
71     }
72     Console.WriteLine(msg);
73     Console.ForegroundColor = ConsoleColor.White;
74     PressEnterToContinue();
75 }
76
77 }
```

```
16 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
62 public static void PrintMessage(string msg, bool success = true, bool pressEnter = true)
63 {
64     if (success)
65     {
66         Console.ForegroundColor = ConsoleColor.Yellow;
67     }
68     else
69     {
70         Console.ForegroundColor = ConsoleColor.Red;
71     }
72     Console.WriteLine(msg);
73     Console.ForegroundColor = ConsoleColor.White;
74     if (pressEnter)
75     {
76         PressEnterToContinue();
77     }
78 }
```

2. Ispravljena greška u liniji 119 u funkciji Convert<T>, tačnije korektno obradjen izuzetak pri konvertovanju.

```
8 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
98 public static T Convert<T>(string prompt)
99 {
100     bool valid = false;
101     string userInput;
102
103     while (!valid)
104     {
105         userInput = GetUserInput(prompt);
106
107         var converter = TypeDescriptor.GetConverter(typeof(T));
108         if (converter != null)
109         {
110             return (T)converter.ConvertFromString(userInput);
111         }
112         else
113         {
114             return default;
115         }
116     }
117
118     return default;
119 }
120 }
```

```

98 8 references | Please sign-in to New Relic CodeStream to see Code Level Metrics
99  public static T Convert<T>(string prompt)
100 {
101     bool valid = false;
102     string userInput;
103
104     while (!valid)
105     {
106         userInput = GetUserInput(prompt);
107
108         try
109         {
110             var converter = TypeDescriptor.GetConverter(typeof(T));
111             if (converter != null)
112             {
113                 return (T)converter.ConvertFromString(userInput);
114             }
115             else
116             {
117                 return default;
118             }
119         }
120         catch
121         {
122             PrintMessage("Invalid input. Try again.", false, false);
123         }
124     }
125     return default;

```

3. Ispravljena greška u liniji 48 u funkciji GetPINInput, tačnije korektno dodana provjera dužine unosa.

```

48 if (inputKey.Key == ConsoleKey.Backspace)
49 {
50     input.Remove(input.Length - 1, 1);
51 }

48 if (inputKey.Key == ConsoleKey.Backspace && input.Length > 0)
49 {
50     input.Remove(input.Length - 1, 1);
51 }

```

Program.cs – Emir Kalajdžija

ATMScreen.cs – Emir Salkić

Utility.cs – Bakir Pljakić

Provjeru ispravljenih grešaka izvršio – **Kerim Halilović**