



TBILISI  
SCHOOL OF  
COMMUNICATION

## Introduction to Programming

### ფინალური გამოცდა N1

პროექტის გამოგზავნის დედლაინი - 23 ოქტომბერი / ოთხშაბათი / 23:59

პროექტის წარდგენა + გასაუბრება - 25 ოქტომბერი / პარასკევი / 9:00-დან

მაქსიმალური ქულა - 100

ფინალური პროექტი - 60 ქულა

წარდგენა - 20 ქულა

გასაუბრება - 20 ქულა

### პროექტის აღწერა

Banking Application

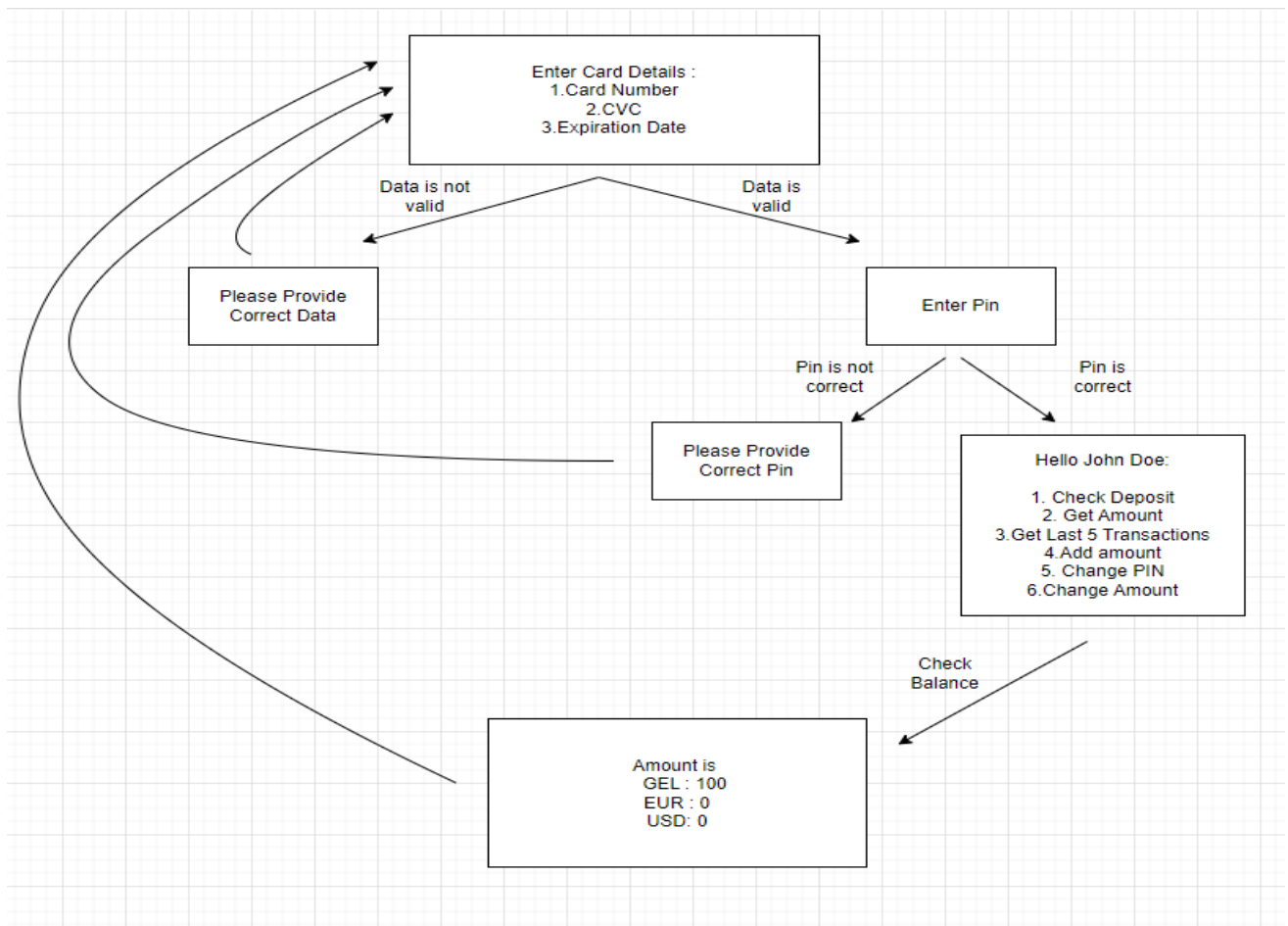
შექმენით ბანკომატის სიმულატორის ტიპის აპლიკაცია .

აპლიკაციამ უნდა შეასრულოს შემდეგი მოქმედებები

- შეამოწმოს ვალიდურია თუ არა ინფორმაცია ( ბარათის ნომერი და ბარათის მოქმედების ვადა)



- შემდეგ ეტაპზე შეამოწმებს პინ კოდს.
- წარუმატებელი ვერიფიკაციის შემთხვევაში აპლიკაცია გამოიტანს შესაბამისს მესიჯს და მოხმარებელს გამოაგდებს სისტემიდან.
- წარმატებული ვერიფიკაციის შემთხვევაში მისცემს საშუალებას აარჩიოს სასურველი მოქმედება
  - ნაშთის ნახვა
  - თანხის გამოტანა ანგარიშიდან
  - ბოლო 5 ოპერაცია
  - თანხის შეტანა ანგარიშზე
  - პინ კოდის შეცვლა
  - ვალუტის კონვერტაცია
- ნებისმიერი ტრანზაქცია შეინახეთ ტრანზაქციების ისტორიის კოლექციაში.
- მოთხოვნის დამუშავების შემდეგ სისტემა უნდა დაბრუნდეს საწყის პოზიციაზე.



## პროექტის ტექნიკური მხარე

- ინფორმაცია შეინახეთ JSON ფაილში. json ფაილის მაგალითი ->

```

{
  "firstName": "John",
  "lastName": "Doe",
  "cardDetails": {
    "cardNumber": "1234-5678-9012-3456",
    "expirationDate": "12/25",
    "CVC": "123"
  },
  "pinCode": "1234",
  "transactionHistory": [
    {
      "transactionDate": "2024-01-03T10:15:30Z",
      "transactionType": "ChangePin",
      "amountGEL": 100,
      "amountUSD": 0,
      "amountEUR": 0
    },
    {
      "transactionDate": "2024-01-03T10:15:30Z",
      "transactionType": "DepositCheck",
      "amount": 100,
      "amountUSD": 0,
      "amountEUR": 0
    },
    {
      "transactionDate": "2024-01-02T15:45:00Z",
      "transactionType": "GetAmount",
      "amount": 500,
      "amountUSD": 0,
      "amountEUR": 0
    },
    {
      "transactionDate": "2024-01-01T08:20:45Z",
      "transactionType": "FillAmount",
      "amount": 600,
      "amountUSD": 0,
      "amountEUR": 0
    },
    {
      "transactionDate": "2024-01-01T08:20:45Z",
      "transactionType": "Balance Inquiry",
      "amount": 0,
      "amountUSD": 0,
      "amountEUR": 0
    }
  ]
}

```

- აპლიკაციამ არ უნდა გამოიტანოს exception. ნებისმიერი Exception უნდა იყოს დაჰენდლილი (Exceptin handling)
- საჭიროა ჩაშენდეს ლოგირების სისტემა (შეგიძლია გამოიყენოთ ბიბლიოთეკები, როგორიცაა [Nlog](#), [GrayLog](#), [SeriLog](#)) ან უბრალოდ ფაილში ჩაწეროთ

## შეფასების კრიტერიუმი

### ტექნიკურად გამართული - 40 ქულა

- რამდენად გამართულად მუშაობს აპლიკაცია
  - აპლიკაცია მუშაობს შეცდომების გარეშე
  - შეცდომის შემთხვევაში გამოაქვს შესაბამისი ტექსტური შეტყობინება და არა exception
  - რამდენად ლოგიკურად და სწორად მუშაობს C# კოდი
- მოდულები სწორად არის შექმნილი

### დალაგებული არქიტექტურა - 20 ქულა

- ფაილური სტრუქტურა რამდენად გამართულია
- რამდენად კარგად არის დალაგებული კოდის არქიტექტურა .

თუ რომელიმე ზემოთ ჩამოთვლილი მითითება არასრულად ან საერთოდ არ არის შესრულებული, ქულა დაგაკლდებათ დაშვებული შეცდომის სირთულის პროპორციულად.

პროექტის შესრულება შესაძლებელია ასევე ფაილების გამოყენების გარეშე ინფომრაციის შესანახად გამოიყენეთ კოლექცია უბრალოდ მიღებულ ქულას დააკლდება 5 ქულა

## პროექტის წარდგენა (ბასაუბრება)

გასაუბრების ფარგლებში შემონმდება რამდენად სწორად ესმის სტუდენტს შესრულებული პროექტი, შესაბამისად, სტუდენტმა უნდა წარადგინოს და ისაუბროს ნამუშევარზე დეტალურად.

- თუ სტუდენტს ესმის მის მიერ დაწერილი კოდი და შეუძლია დეტალურად ახსნას აპლიკაციის-ს შექმნის თითოეული ეტაპი ასევე შეუძლია მის მიერ შექმნის აპლიკაციისთან მუშაობა - 25-40 ქულა
- სტუდენტი ავლენს ზედაპირულ ცოდნას და გაპარულია რამდენიმე ნიუანსი - 10-25 ქულა
- თუ სტუდენტმა დააკოპირა სხვისი კოდი და ვერ საუბრობს შესრულებულ დავალებაზე - 0-10 ქულა
- თუ სტუდენტს არ აქვს გაკეთებული პროექტი, მაშინ ის ავტომატურად იღებს 0 ქულას.

## ბასაუბრება

## **1 ამოცანა**

თითოეული ამოცანა შეფასდება 20 ქულით. 5 წუთი იქნება განკუთვნილი თითო ამოცანის ამოხსნაზე.