

# BackEnd-ის (C# / Python) პირველი ეტაპის შემაჯამებელი პროექტი

1. BackEnd-ის პირველი ეტაპის დასრულების შემდგე სტუდენტს გავლილი აქვს C# / Python-ის პროგრამირების ენის მოდული. შესაბამისად მან უნდა წარმოადგინოს C# / Python -ის კონსოლ აპლიკაციის პროექტი.

2. C# / Python -ის კონსოლ აპლიკაციების მინი-პროექტების ჩამონათვალიდან ლექტორი სტუდენტებს უნაწილებს პროექტების 2-3 სხვადასხვა ნაკრებს.

2.1 C#-ის კონსოლ აპლიკაცია სტუდენტს ჩაეთვლება შესრულებულად თუ ის აკმაყოფილებს შემდეგ კრიტერიუმებს:

- 2.1.1 პროექტში გამოყენებული უნდა იყოს სხვადასხვა ტიპის ცვლადები
- 2.1.2 პროექტში გამოყენებული უნდა იყოს პირობითი და ციფლის ოპერატორები
- 2.1.3 პროექტში გამოყენებული უნდა იყოს მასივები
- 2.1.4 პროექტში სტუდენტს შექმნილი უნდა ჰქონდეს სამომხმარებლო ფუნქციები
- 2.1.5 სტუდენტმა უნდა გამოამჟღავნოს ფაილების შექმნის/რედაქტირების/წაშლის უნარები.
- 2.1.6 პროექტში სტუდენტმა უნდა გამოამჟღავნოს ოპა-ის პრინციპების ცოდნა (მემკვიდრეობა, ინკაფსულაცია, პოლიმორფიზმი).
- 2.1.7 კოდის ყველა მოდული დაკომენტარებული უნდა იყოს.

3. ერთი და იგივე პროექტის შესრულებისას შესაძლებელია 2-3 კაციანი ქვეაგუფების შექმნა.

4. სტუდენტმა შესრულებული პროექტების კოდები უნდა ატვირტოს საკუთარ პერსონალურ GitHub-ის რეპოზიტორიაში, და საჯარო ლინკი გაუზიაროს ლექტორს. პროექტის ლინკი წარმოადგენს პროექტის შესრულების დამატებით დიკუმენტს.

=====

# კონსოლ აპლიკაციის დოკუმენტაცია

ამ დოკუმენტაციაში წარმოდგენილია ბეჭედზე პროგრამირების მიმართულყების შუალედური კონსოლ აპლიკაციების პროექტების აღწერები:

- კალკულატორი,
- თამაში - გამოიცანი რიცხვი
- თამაში - Hangman
- თარჯიმანი
- ბანკომატი

## სარჩევი

- შესავალი
- ფუნქციონალობა
- დამატებითი ინფორმაცია

## 1. შესავალი

კონსოლ აპლიკაციები არის C# / Python პროექტი, რომელიც აერთიანებს კალკულატორს, ორ თამაშს (გამოიცანი რიცხვი და Hangman), თარჯიმანის და ბანკომატის ფუნქციებს. ამ აპლიკაციის მიზანია პროგრამირების ენების C# / Python მოდულის გავლის შემდეგ სტუდენტმა მოახდინოს მიღებული ცოდნის დემონსტრირება.

## 2. ფუნქციონალობა

კონსოლ აპლიკაცია მოიცავს შემდეგ ფუნქციონალობას:

### 2.1 კალკულატორი

კალკულატორის ფუნქცია მომხმარებლებს საშუალებას აძლევს შეასრულონ ძირითადი არითმეტიკული მოქმედებები (+, -, \*, /). მომხმარებლებს შეუძლიათ შეიყვანონ ორი რიცხვი და შემდეგ აირჩიონ ოპერაცია შედეგის მისაღებად. კალკულატორი ასევე შეიცავს შეყვანის ვალიდაციას არასწორი შეყვანების დასამუშავებლად.

## **2.2 თამაში 1: გამოიცანით რიცხვი**

ამ თამაშში პროგრამა აგენერირებს შემთხვევით რიცხვს მითითებული დიაპაზონიდან. მომხმარებლებს სთხოვენ გამოიცნონ რიცხვი. არასწორი რიცხვის შემთხვევაში პროგრამა მომხმარებელს აძლევს მინიშნებას (უფრო მაღალი/უფრო დაბალი). თამაში აკონტროლებს მცდელობების რაოდენობას და აჩვენებს შედეგს, როდესაც მომხმარებელი გამოიცნობს სწორ რიცხვს.

## **2.3 თამაში 2: Hangman**

Hangman არის სიტყვაშის გამოცნობის თამაში. პროგრამა ირჩევს შემთხვევით სიტყვას წინასწარ განსაზღვრული სიიდან და აჩვენებს მას ქვედა ტირეების გამოყენებით (რამდენი ასოცაა სიტყვაში, იმდენი ქვედა ტირე), რომელიც წარმოადგენს ფარულ ასოებს. მომხმარებლებს სთხოვენ გამოიცნონ ასო და პროგრამა ამოწმებს არის თუ არა ასო სიტყვაში. ვლინდება სწორად გამოცნობილი ასოები და თამაში გრძელდება მანამ, სანამ მომხმარებელი არ გამოიცნობს სიტყვას ან არ ამოიწურება მცდელობები.

## **2.4 თარჯიმანი**

თარჯიმანის აპლიკაცია მომხმარებელს საშუალებას აძლევს აირჩიოს ენების წყვილი - რომელი ენიდან რომელ ენეაზე უნდა ითარგმნოს სიტყვები (მაგ: ქართული-ინგლისური, ქართული-რუსული, ინგლისური-ქართული, რუსული-ქართული). ამის შემდეგ მომხმარებელს შეყავს სიტყვა ან მოკლე ფრაზა. პროგრამა კითხულობს თარგმანებს ტექსტური ფაილიდან (ლექსიკონიდან) და აჩვენებს შესაბამის ნათარგმნ სიტყვას ან ფრაზას. იმ შემთხვევაში თუ სათარგმნი სიტყვა არ აღმოჩნდა ლექსიკონში, პროგრამა მომხმარებელს სთავაზობს დაამატოს ამ ახალი სიტყვის თარგმანი ლექსიკონს (ტექსტურ ფაილს).

## **3.5 ბანკომატი**

ბანკომატის აპლიკაცია საშუალებას აძლევს მომხმარებელს განახორციელოს ბანკომატის ძირითადი ოპერაციების სიმულაცია. მომხმარებელს შეუძლია შესრულოს ისეთი ქმედებები, როგორიცაა:

- რეგისტრაცია,
- დალოგინება,
- ბალანსის შემოწმება,
- თანხის შეტანა და
- თანხის გატანა,
- მომხმარებლის მონაცემები ინახება ტექსტურ ფაილში მომხმარებლის ანგარიშის ინფორმაციის შესანარჩუნებლად.

სტუდენტს შეუძლია უფრო გაართულოს ფუნქციონალი შემდეგი პირობის შესრულებბის გზით:

- სესხის მოთხოვნა და
- მოთხოვნის დადასტურება ადმინის მხრიდან (იგულისხება, რომ ამ შემთხვევაში რეგისტრაციის დროს მომხმარებელს აქვს 2 როლი: client, admin),

## 4. დაამატებითი ინფორმაცია

- წარმოდგენილი აღწერა წარმოადგენს მინიმალურ მოთხოვნებს კონსოლ აპლიკაციისთვის. თქვენ შეგიძლიათ შეცვალოთ და გაართულოთ ის თქვენი მოსაზრებებისამებრ.
- აპლიკაციის გამოყენებისას უზრუნველყოფილი უნდა იყოს შეყვანის სათანადო ვალიდაცია და შეცდომების დამუშავება.
- თარჯიმანის ფუნქციისთვის თარგმანები იყითხება ტექსტური ფაილიდან. შეცვალეთ ტექსტური ფაილი, რომ დაამატოთ ან შეცვალოთ თარგმანები საჭიროებისამებრ.
- ბანკომატის ფუნქცია ინახავს მომხმარებლის ანგარიშის მონაცემებს ტექსტურ ფაილში. შეცვალეთ ტექსტური ფაილი მომხმარებლის ანგარიშებისა და მათთან დაკავშირებული ინფორმაციის სამართავად.

---

## ოთვ პროექტები

---

**წიგნების მართვის კონსოლ აპლიკაცია**

შექმენით კონსოლ აპლიკაცია C# / Python-ში, რომელიც მომხმარებლებს საშუალებას აძლევს მართონ წიგნების სია.

თითოეულ წიგნს აქვს სათაური, ავტორი და გამოცემის წელი. მომხმარებელს უნდა შეეძლოს წიგნის დამატება, წიგნების სიის ნახვა და წიგნის სათაურის მოძიება.

განახორციელეთ შემდეგი ფუნქციები ობიექტზე ორიენტირებული პროგრამირების პრინციპების გამოყენებით:

## 1. Book კლასი:

- შექმენით წიგნის კლასი შემდეგი თვისებებით:
  - სათაური
  - ავტორი
  - გამოცემის წელი

## 2. BookManager კლასი:

- შექმენით BookManager კლასი წიგნების სიის სამართავად.
- კლასი უნდა შეიცავდეს მეთოდებს:
  - ახალი წიგნის დამატება სიაში.
  - ყველა წიგნის სიის ჩვენება.
  - წიგნის ძებნა მისი სათაურის მიხედვით.

## 3. User Interface:

- კონსოლზე დაფუძნებული მომხმარებლის ინტერფეისის შემუშავება, რომელიც მომხმარებლებს საშუალებას აძლევს:
  - დაამატოს ახალი წიგნი მისი სათაურის, ავტორისა და გამოცემის წლის შეყვანით.
  - იხილეთ ყველა წიგნის დეტალური ინფორმაციის ნახვა.
  - წიგნის ძებნა მისი სათაურის შეყვანის გზით.

## 4. Validation

- განახორციელეთ შეყვანის ვალიდაცია, რათა დარწმუნდეთ, რომ მომხმარებელი შეიყვანს ვალიდურ ინფორმაციას.

## 5. მაგალითი:

მოიყვანეთ მაგალითი იმის სადემონსტრაციოდ, თუ როგორ მუშაობს აპლიკაცია წიგნების დამატების, სიის დათვალიერებისა და წიგნის ძიების პროცესში.

**შენიშვნა:** გამოიყენეთ ობიექტზე ორიენტირებული პროგრამირების ცნებები, როგორიცაა კლასები და ობიექტები თქვენი კოდის სტრუქტურირებისთვის. ყურადღება მიაქციეთ ინკაფსულაციას, მემკვიდრეობას და პოლიმორფიზმს, სადაც ეს შესაძლებელია.

## სტუდენტების მართვის სისტემა

### აღწერა:

შექმენით კონსოლის აპლიკაცია C# / Python-ში, რომელიც წარმოადგენს სტუდენტის მართვის მარტივ სისტემას ობიექტზე ორიენტირებული პროგრამირების პრინციპების გამოყენებით. სისტემამ მომხმარებელს უნდა მისცეს საშუალება განახორციელოს ძირითადი ოპერაციები, რომლებიც დაკავშირებულია სტუდენტის ინფორმაციასთან. თითოეულ სტუდენტს უნდა ჰქონდეს ისეთი ატრიბუტები, როგორიცაა სახელი, სიის ნომერი და შეფასება.

### მოთხოვნები:

#### 1. სტუდენტური კლასი:

- შექმენით სტუდენტური კლასი შემდეგი ატრიბუტებით:
  - Name Name (string) - სახელი
  - Roll Number (int) - სიის ნომერი
  - Grade (char) - შეფასება.

## **2. კონსოლის აპლიკაცია:**

- კონსოლის აპლიკაციის შემუშავება რომელიც მომხმარებელს მისცემს საშუალებას იურთიერთოს სტუდენტთა მართვის სისტემასთან.

## **3. მენიუს სისტემა:**

- მენიუს ჩვენება შემდეგი პარამეტრებით:
  - ახალი სტუდენტის დამატება
  - ყველა სტუდენტის ნახვა
  - სტუდენტის ძებნა ნომრის მიხედვით
  - მოსწავლის შეფასების განახლება
  - გასვლა

## **4. ფუნქციონალობა:**

### **• ახალი სტუდენტის დამატება:**

- სთხოვეთ მომხმარებელს შეიყვანოს მოსწავლის სახელი, სიის ნომერი და შეფასება.
- შექმნით ახალი სტუდენტური ობიექტი და დაამატეთ ის სტუდენტების სიაში.

### **• ყველა სტუდენტის ნახვა:**

- სისტემაში ყველა სტუდენტის დეტალების ჩვენება.

### **• მოძებნეთ სტუდენტი ნომრის მიხედვით:**

- შესთავაზეთ მომხმარებელს შეიყვანოს სიის ნომერი.
- მოძებნეთ სტუდენტი მითითებული ნომრით და აჩვენეთ მათი დეტალები.

### **• განაახლეთ მოსწავლის შეფასება:**

- შესთავაზეთ მომხმარებელს შეიყვანოს იმ სტუდენტის სიის ნომერი, რომლის შეფასებაც განაახლებას საჭიროებს.
- ნება მიეცით მომხმარებელს განაახლოს შეფასება მითითებული სტუდენტისთვის.

- გასვლა:
  - შეწყვიტე პროგრამა.

## 5. ობიექტზე ორიენტირებული პრინციპები:

- საჭიროების შემთხვევაში გამოიყენეთ ინკაფუსულაცია, მემკვიდრეობა და პოლიმორფიზმი.

## 6. ვალიდაცია

- განახორციელეთ შეყვანის ვალიდაცია, რათა დარწმუნდეთ, რომ მომხმარებელი შეიყვანს ვალიდურ ინფორმაციას.

**შენიშვნა:** გამოიყენეთ ობიექტზე ორიენტირებული პროგრამირების ცნებები, როგორიცაა კლასები და ობიექტები თქვენი კოდის სტრუქტურირებისთვის. ყურადღება მიაქციეთ ინკაფუსულაციას, მემკვიდრეობას და პოლიმორფიზმს, სადაც ეს შესაძლებელია.

---