

Zadanie egzaminacyjne

Wykonaj aplikację konsolową oraz mobilną według wskazanych wymagań. Wykonaj dokumentację zgodnie z opisem w części III instrukcji do zadania. Wykorzystaj konto Egzamin bez hasła.

1. **Stwórz folder o nazwie zgodnej z numerem zdającego.** W folderze utwórz podfoldery: konsolowa, mobilna, dokumentacja. Po wykonaniu każdej aplikacji, jej pełny kod (cały folder projektu) spakuj do archiwum. Następnie pozostaw w podfolderze jedynie spakowane archiwum, pliki źródłowe modyfikowane przez Ciebie oraz, jeżeli istnieje, plik wykonywalny.
-

Część I. Aplikacja konsolowa

1. **Napisz program obliczający ciąg Fibonacciego.** Program powinien:
 - o Pobierać liczbę całkowitą n określającą ilość liczb w ciągu Fibonacciego do obliczenia ($n \geq 2$).
 - o Obliczać i wypisywać kolejne liczby ciągu Fibonacciego do liczby n .
 - o Na końcu wyświetlać sumę wszystkich liczb w ciągu do liczby n .
2. **Działanie programu:**
 - o Program pyta użytkownika o liczbę n . Jeśli podana wartość nie jest liczbą całkowitą lub jest mniejsza niż 2, program wyświetla odpowiedni komunikat błędu i prosi o ponowny wprowadzenie.
 - o Po obliczeniu ciągu program wyświetla wynik w formacie: Liczba <numer>: <wartość>, oraz sumę ciągu.
3. Definicja ciągu Fibonacciego: zerowy numer ciągu wynosi 0, pierwszy wynosi 1, a każdy następny otrzymujemy dodając dwa poprzednie:

$$F_n = \begin{cases} 0 & , \text{ dla } n = 0 \\ 1 & , \text{ dla } n = 1 \\ F_{n-2} + F_{n-1} & , \text{ dla } n > 1 \end{cases}$$

Kilka kolejnych wyrazów tego ciągu według pierwszej definicji przedstawia się następująco:

1,1,2,3,5,8,13,21,...

Założenia do programu:

- Program powinien być napisany w jednym z języków: C++, C#, Java lub Python.
- Wynik ciągu powinien być przechowywany w tablicy lub kolekcji.
- Wprowadzanie i sprawdzanie poprawności danych zaimplementowane w osobnej funkcji zwracająca typ całkowity

- Obliczanie ciągu Fibonacciego i umieszczenie go w tablicy lub kolekcji zaimplementowane w osobnej funkcji zwracająca kolekcję, przyjmująca wartość całkowitą.
- Program powinien być czytelnie sformatowany, ze zrozumiałą komunikacją dla użytkownika i komentarzami do kluczowych funkcji.
- Dokumentację aplikacji wykonaj zgodnie z częścią III zadania.

Część II. Aplikacja mobilna

1. **Wykonaj aplikację mobilną** przedstawiającą interaktywnie wyliczony ciąg Fibonacciego:
 - Aplikacja powinna zawierać pole wejściowe na liczbę n oraz przycisk „Oblicz ciąg”.
 - Po kliknięciu „Oblicz ciąg” aplikacja powinna wyświetlić kolejne liczby ciągu Fibonacciego do n oraz ich sumę.
 - Powinna być także możliwość zresetowania wyników.
2. **Elementy aplikacji:**
 - Pole do wprowadzenia liczby n.
 - Przycisk „Oblicz ciąg”.
 - Przycisk „Resetuj”.
 - Obszar wyświetlający kolejne liczby ciągu i ich sumę.
 - Etykieta tytułowa z treścią „Ciąg Fibonacciego. Autor: <numer zdającego>”.
3. **Założenia aplikacji:**
 - Interfejs użytkownika zapisany za pomocą języka znaczników (XAML, XML) odpowiedniego dla wybranego środowiska.
 - Kolory tła, rozmieszczenie elementów i stylizacja zgodna z opisem zadania.
 - Nazwy zmiennych i metod w kodzie źródłowym powinny być zrozumiałe i adekwatne do ich funkcji.

Część III. Dokumentacja aplikacji konsolowej i mobilnej

1. **Utwórz dokumentację w kodzie aplikacji konsolowej** – dodaj nagłówek wybranej funkcji, opisując jej działanie, parametry oraz wynik zwracany według poniższego wzoru:

```
*****
nazwa: <nazwa funkcji / metody>
opis: <opis działania funkcji / metody>
parametry: <opis parametru1, lub „brak”>
<opis parametru2, lub „brak”>
zwracany typ i opis: <nazwa typu i opis>
autor: <numer zdającego>
*****
```

2. **Zrzuty ekranu dokumentujące uruchomienie aplikacji:**

- Aplikacja konsolowa: przedstaw kolejne kroki programu, zapisując zrzuty jako konsola1, konsola2, itd.
 - Aplikacja mobilna: zapisz zrzuty po uruchomieniu programu, wynikach obliczeń i po użyciu przycisku „Resetuj”.
3. **Plik z dokumentacją:** w edytorze tekstu utwórz plik egzamin zawierający:
- Nazwę systemu operacyjnego, na którym pracował zdający.
 - Nazwy środowisk programistycznych i emulatorów używanych podczas egzaminu.
 - Nazwy języków programowania użytych do realizacji zadania.