

Część 1. Aplikacja konsolowa.

Wykonaj aplikację konsolową za pomocą środowiska programistycznego dostępnego na stanowisku. Aplikacja służy do sprawdzania czy sportowiec (biegacz) biegnąc na dany dystans nie pobił przypadkiem rekordu na mniejszy dystans (np. biegnąc na 10km można pobić rekord na 3km, czy 5km). Napisz program, który pobiera czasy użytkownika (w sekundach) na każdym kilometrze i wylicza jego czas w odcinku o podanej wielkości.

Przykład działania:

Ile kilometrów przebiegłeś: 10

Podaj czasy na każdym kilometrze:

140

230

260

240

270

200

170

150

210

220

Podaj jakiej wielkości najlepszego odcinka szukasz (w km): 3

(Odpowiedz programu)

Najlepszy czas na 3km odcinku wynosi: 520

Wyjaśnienie:

Najlepszy czas na odcinku 3km składa się z czasów 200+170+150 sekund, co daje 520s.

Zwróć uwagę, że najlepszy czas na odcinku 1km to 140, ale nie jego szukamy, tylko takich trzech kolejnych czasów, które są równe długości odcinka o podanej wielkości (3km).

Założenia do programu:

- Obiektowy język programowania zgodny z zainstalowanym na stanowisku: C++ lub C# lub Java lub Python.

- Program działa w konsoli

- Program jest odporny na błędy wejścia, czyli jeśli ktoś poda np. tekst zamiast liczby to program wyświetli błąd i zakończy pracę

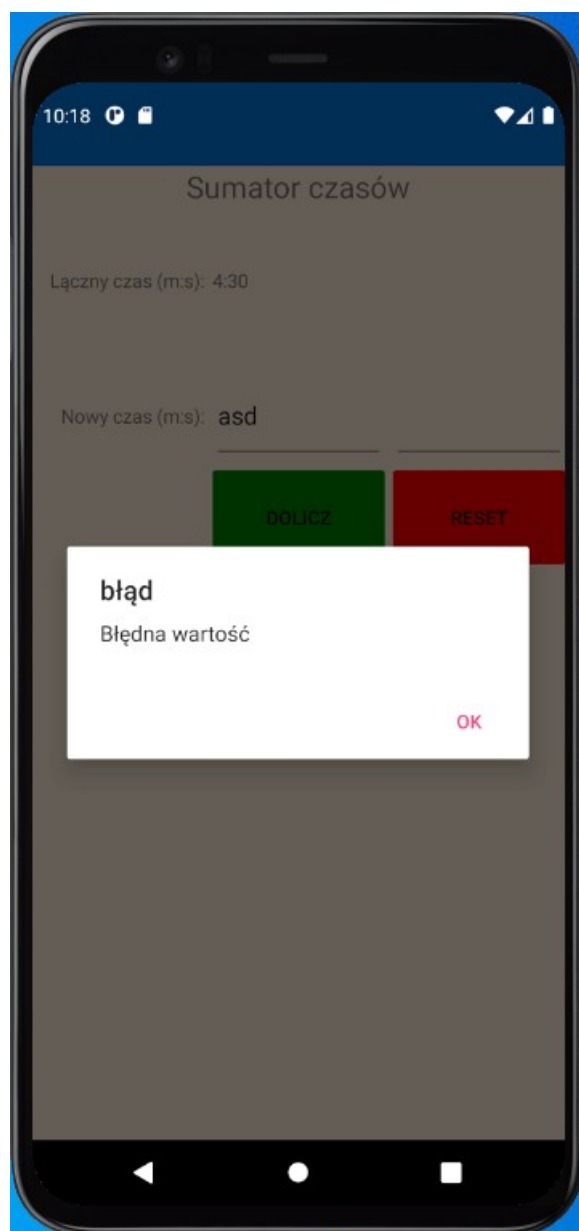
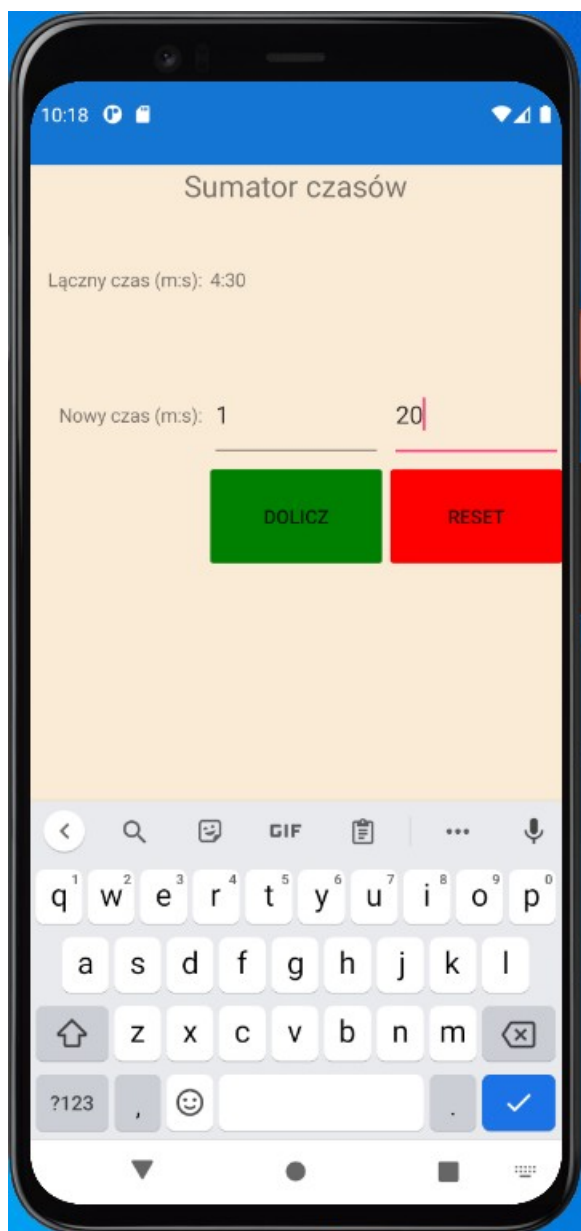
- Program powinien być zapisany czytelnie, z zasadami czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji.

- Punktowana jest też złożoność obliczeniowa (czyli program ma działać w czasie krótszym niż $O(n^2)$)

- Dokumentacja do programu wykonana zgodnie z wytycznymi z części III zadania

Część II. Aplikacja mobilna

Wykonaj aplikację mobilną za pomocą środowiska programistycznego dostępnego na stanowisku oraz uruchom ją w dostępnym emulatorze systemu mobilnego. Aplikacja służy do sumowania czasów z kolejnych odcinków (lub okrążeń). Jej zadaniem jest zliczenie czasów z każdego odcinka (okrążeń) i wypisaniu sumy w postaci minuty:sekundy



Na rysunku po lewej stan działania aplikacji po wpisaniu czasu 2:50 oraz 1:40 (dało to łączny czas 4:30), oraz przed dodaniem czasu 1:20 (jeszcze nie wciśnięto trzeci raz przycisku „Dolicz”).
Na rysunku po prawej stronie efekt wpisania błędnej wartości i naciśnięcia przycisku „Dolicz”

Elementy aplikacji:

- tytuł „Sumator czasów” (wyśrodkowany, u góry)
- tekst „Łączny czas (m:s): 0:00”
- tekst „Nowy czas (m:s):”
- pole do wpisywania minut
- pole do wpisywania sekund
- przycisk „Dolicz”
- przycisk „Reset” lub „Resetuj”

Działanie aplikacji:

- użytkownik podaje czas kolejnego odcinka (okrążenia) w polach do wprowadzania minut oraz sekund
- naciśnięcie klawisza „Dolicz” spowoduje dodanie nowego czasu do aktualnej sumy i zaktualizowanie na ekranie tej sumy („Łączny czas”). Jednocześnie pola do wprowadzania minut i sekund stają się pustą (gotowe na kolejne dane)
- naciśnięcie klawisza „Reset” powoduje wyczyszczenie pól do wprowadzania danych oraz ustawienie łącznego czasu na 0:00

Założenia aplikacji:

- Interfejs użytkownika zapisany za pomocą języka znaczników wspieranego w danym środowisku (np. XAML, XML).
- tytuł „Sumator czasów” ma być wyśrodkowany i używać dużej czcionki
- pomiędzy napisem „Łączny czas (m:s):” a czasem obok nie powinno być wyraźnego odstępu
- pomiędzy napisem „Nowy czas (m:s)” a polem tekstowym obok nie powinno być wyraźnego odstępu
- sekundnik ma zawsze wypisywać wartość dwu cyfrową (np. 6 sekund to „06”)
- tło ma być w jasnym kolorze, ale nie białym
- przycisk z napisem „Dolicz” ma być zielony
- przycisk z napisem „Reset” ma być czerwony
- naciśnięcie przycisku „Dolicz” spowoduje doliczenie wpisanego czasu do łącznego czasu (pole tekstowe powinno się zaktualizować) oraz wyczyszczenie pól tekstowych
- naciśnięcie przycisku „Reset” powinno ustawić łączny czas na 0:00 oraz wyczyścić pola tekstowe
- program powinien być odporny na wpisywanie wartości nie numerycznych w polach tekstowych. Wpisanie takiej wartości i naciśnięcie przycisku „Dolicz” powinno wyświetlić alerta z komunikatem „Błędna wartość”
- program powinien być odporny na wpisanie nieprawidłowej wartości sekund (np. 70, czy -5). Wpisanie takiej wartości i naciśnięcie przycisku „Dolicz” powinno wyświetlić alerta z komunikatem „Błędna wartość”
- Program powinien być zapisany czytelnie, z zasadami czystego formatowania kodu, należy stosować znaczące nazwy zmiennych i funkcji.
- Dokumentacja do programu wykonana zgodnie z wytycznymi z części III zadania

Część III. Dokumentacja utworzonych aplikacji

W aplikacji konsolowej funkcję/metodę znajdującą długość szukanego odcinka należy napisać odpowiedni komentarz zgodnie z podanym szablonem:

```
/**
 * <NazwaMetody>
 * <Krótki opis co metoda robi>
 * Parametry wejściowe:
 * <parametr 1> - <opis co to za parametr>
 * <parametr 2> - <opis co to za parametr>
 * ...
 * wartość zwracana:
 * <opis co zwraca metoda. Lub jakie pola klasy modyfikuje>
 **/
```

Jeżeli masz więcej funkcji/metod niż jedna to też je udokumentuj w ten sam sposób.

Wykonaj zrzuty ekranu dokumentujące uruchomienie aplikacji. Zrzuty powinny obejmować cały obszar ekranu z widocznym paskiem zadań. Jeżeli aplikacja uruchamia się, na zrzucie należy umieścić okno z wynikiem działania programu oraz otwarte środowisko programistyczne z projektem. Jeżeli aplikacja nie uruchamia się z powodu błędów kompilacji, należy na zrzucie umieścić okno ze spisem błędów i widocznym otwartym środowiskiem programistycznym.

Wymagane zrzuty ekranu:

– z aplikacji konsolowej – *konsola.jpg*

– z aplikacji mobilnej – *mobilna.jpg*

- z aplikacji mobilnej z komunikatem błędu na ekranie (użytkownik wprowadził błędne dane) – *mobilna_błąd.jpg*

opracował: Marcin Gardyjan