

Analiza pomiarów tętna

Problem

Monitorowanie stanu zdrowia jest ważne, jednym z jego elementów jest analiza pomiarów tętna. Tętno inaczej nazywane jest też pulsem. Istnieje wiele urządzeń, które pozwalają na automatyczny pomiar tętna (pulsometry, pulsoksymetry, opaski, zegarki z pulsometrem). Dla większej liczby zebranych pomiarów tętna często wyznacza się tętno minimalne, maksymalne i średnie. Ta informacja pozwala wykryć anomalie w pracy serca.

Ćwiczenie

Proszę napisać program, który prosi użytkownika o wprowadzenie serii pomiarów tętna a następnie poinformuje go jakie było tętno średnie, minimalne i maksymalne. Liczba pomiarów wartości tętna nie jest z góry znana, program powinien najpierw zapytać o liczbę pomiarów a potem wczytać kolejne pomiary. Proszę rozważyć zabezpieczenie programu przed wprowadzeniem nieprawidłowej liczby pomiarów i samego pomiaru tętna. Wartość tętna jest liczbą całkowitą.

Testy

Proszę przetestować program wprowadzając wybrane różne, w tym nieprawidłowe wartości liczby pomiarów, różne wartości pomiarów tętna i wprowadzić poprawność wyznaczania wartości średniej, minimum i maksimum.

Przykładowe rozwiązania – wersja pierwsza

Język Python

Wersja 1

```
import sys

pomiarPulsu = 0
numerPomiaru = 0
sumaPomiarow = 0
maksymalnyPomiar = 0
minimalnyPomiar = sys.maxsize
```

```

print("Analizuję serię pomiarów tętna")
liczbaPomiarow = int(input("Ile będzie pomiarów: "))

print("Podaj kolejne pomiary")
while numerPomiaru < liczbaPomiarow:
    pomiarPulsu = int(input(f"{numerPomiaru + 1: 3}: "))
    sumaPomiarow = sumaPomiarow + pomiarPulsu
    if pomiarPulsu > maksymalnyPomiar:
        maksymalnyPomiar = pomiarPulsu
    if pomiarPulsu < minimalnyPomiar:
        minimalnyPomiar = pomiarPulsu
    numerPomiaru = numerPomiaru + 1

print("Średnie tętno:", sumaPomiarow / liczbaPomiarow)
print("Maksymalne tętno:", maksymalnyPomiar)
print("Minimalne tętno:", minimalnyPomiar)

```

Wersja 2 - kontrola liczby wprowadzanych pomiarów

W przypadku wprowadzenia przez użytkownika napisu nie będącego prawidłową liczbą całkowitą, wygenerowany zostanie wyjątek przez funkcję *int*. Instrukcja *try-except* pozwala na obsługę tego wyjątku. Została ona osadzona w instrukcji iteracyjnej *while*, która stała się pętlą poprzez wstawienie wartości *True* do warunku iteracji. Instrukcja *if* oraz *break* pozwalają na opuszczenie pętli w przypadku wprowadzenia wartości prawidłowej. W przypadku wprowadzenia wartości mniejszej od zera, zerowej lub nieprawidłowej liczby, program zapyta ponownie o wartość liczby pomiarów.

```

import sys

pomiarPulsu = 0
numerPomiaru = 0
sumaPomiarow = 0
maksymalnyPomiar = 0
minimalnyPomiar = sys.maxsize

print("Analizuję serię pomiarów tętna")
while True:
    try:
        liczbaPomiarow = int(input("Ile będzie pomiarów: "))
        if liczbaPomiarow > 0:
            break
        else:
            print("Liczba pomiarów powinna być większa od zera")
    except ValueError:
        print("Błędna liczba")

```

```

print("Podaj kolejne pomiary")
while numerPomiaru < liczbaPomiarow:
    pomiarPulsu = int(input(f"{numerPomiaru + 1: 3}: "))
    sumaPomiarow = sumaPomiarow + pomiarPulsu
    if pomiarPulsu > maksymalnyPomiar:
        maksymalnyPomiar = pomiarPulsu
    if pomiarPulsu < minimalnyPomiar:
        minimalnyPomiar = pomiarPulsu
    numerPomiaru = numerPomiaru + 1

print("Średnie tętno:", sumaPomiarow / liczbaPomiarow)
print("Maksymalne tętno:", maksymalnyPomiar)
print("Minimalne tętno:", minimalnyPomiar)

```

Język Java

Wersja 1

```

import java.util.Scanner;

public class Main
{
    public static void main(String[] args)
    {
        int sumaPomiarow = 0;
        int numerPomiaru = 0;
        int liczbaPomiarow = 0;
        int pomiarPulsu = 0;
        int maksymalnyPomiar = 0;
        int minimalnyPomiar = Integer.MAX_VALUE;

        Scanner wejscie = new Scanner(System.in);
        System.out.println("Analizuję serię pomiarów tętna");

        System.out.print("Ile będzie pomiarów: ");
        liczbaPomiarow = wejscie.nextInt();

        while(numerPomiaru < liczbaPomiarow)
        {
            System.out.printf("%2d: ", numerPomiaru + 1);
            pomiarPulsu = wejscie.nextInt();
            sumaPomiarow += pomiarPulsu;
            if(pomiarPulsu > maksymalnyPomiar)
                maksymalnyPomiar = pomiarPulsu;
            if(pomiarPulsu < minimalnyPomiar)
                minimalnyPomiar = pomiarPulsu;
            ++numerPomiaru;
        }
        System.out.printf("Średni puls: %.2f\n",

```

```

        (float)sumaPomiarow / liczbaPomiarow);
    System.out.printf("Maksymalny puls: %d\n", maksymalnyPomiar);
    System.out.printf("Minimalny puls: %d\n", minimalnyPomiar);
}
}

```

Wersja 2 - kontrola liczby wprowadzanych pomiarów

W przypadku wprowadzenia przez użytkownika napisu nie będącego prawidłową liczbą całkowitą, wygenerowany zostanie wyjątek przez metodę *nextInt*. Instrukcja *try-catch* pozwala na obsługę tego wyjątku. Uwaga, nieprawidłowy ciąg znaków pozostanie nieodczytany w buforze klawiatury, dlatego we frazie *catch* następuje puste czytanie aktualnej zawartości bufora z wykorzystaniem metody *next*. Całość została osadzona w instrukcji iteracyjnej *while*, która stała się pętlą poprzez wstawienie wartości *true* do warunku iteracji. Instrukcja *if* oraz *break* pozwalają na opuszczenie pętli w przypadku wprowadzenia wartości prawidłowej. W przypadku wprowadzenia wartości mniejszej od zera, zerowej lub nieprawidłowej liczby, program zapyta ponownie o wartość liczby pomiarów.

```

import java.util.Scanner;

public class Main
{
    public static void main(String[] args)
    {
        int sumaPomiarow = 0;
        int numerPomiaru = 0;
        int liczbaPomiarow = 0;
        int pomiarPulsu = 0;
        int maksymalnyPomiar = 0;
        int minimalnyPomiar = Integer.MAX_VALUE;

        Scanner wejscie = new Scanner(System.in);
        System.out.println("Analizuję serię pomiarów tętna");

        while(true)
            try
            {
                System.out.print("Ile będzie pomiarów: ");
                liczbaPomiarow = wejscie.nextInt();
                if(liczbaPomiarow > 0)
                    break;
                else
                    System.out.println("Liczba pomiarów powinna być większa \
od zera");
            }
            catch(Exception e)
            {

```

```

        System.out.println("Błędna liczba");
        wejscie.next(); // Puste czytanie, czyszczenie bufora
    }

    while(numerPomiaru < liczbaPomiarow)
    {
        System.out.printf("%2d: ", numerPomiaru + 1);
        pomiarPulsu = wejscie.nextInt();
        sumaPomiarow += pomiarPulsu;
        if(pomiarPulsu > maksymalnyPomiar)
            maksymalnyPomiar = pomiarPulsu;
        if(pomiarPulsu < minimalnyPomiar)
            minimalnyPomiar = pomiarPulsu;
        ++numerPomiaru;
    }
    System.out.printf("Średni puls: %.2f\n",
        (float)sumaPomiarow / liczbaPomiarow);
    System.out.printf("Maksymalny puls: %d\n", maksymalnyPomiar);
    System.out.printf("Minimalny puls: %d\n", minimalnyPomiar);
}
}

```

Rozszerzenie

Proszę rozwinąć program o wyznaczenie ile we wprowadzonej serii pomiarów było za niskich wartości tętna (wartość mniejsza od 60), prawidłowych (wartości pomiędzy 60 a 100), zbyt wysokich (wartość większa od 100). Te informacje mają zostać przedstawione użytkownikowi programu.