

# Średniozaawansowane programowanie w C++

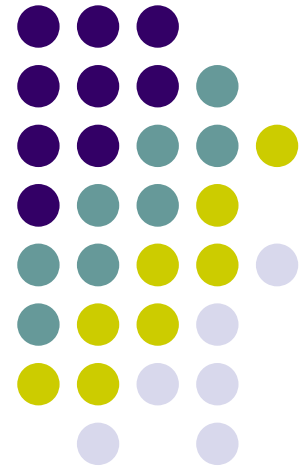
---

## **Program #1**

21 października 2020 r.

na

4 listopada 2020 r.



# Zadanie



Napisać program dla kontrwywiadu analizujący billingi telefoniczne tajnych agentów Wrogich Mocarstw. Przykładowy billing (zawierający kolejno: dzień miesiąca, numer telefonu, czas trwania połączenia [s]) przechwycony przez zaprzyjaźnione służby znaleźć można na stronie. Program winien czytać plik tekstowy z billingiem ze standardowego wejścia oraz wyrzucać opracowaną statystykę na standardowe wyjście. Oczekuje się, że raport będzie sporządzony wg wzorca.

W ogólności nie jest znany zbiór krajów, do których dzwonił agent, ani długość numerów, a tym bardziej liczba wykonanych połączeń. Można natomiast założyć, że wszystkie kody krajów są dwucyfrowe.

# Podział na jednostki kompilacji



## 1. main - bardzo krótki!

Wywołanie konstruktora klasy Billing (wczytanie danych z pliku)  
Wywołanie metod liczących statystyki i drukujących na ekran

## 2. histogram

class Histogram - liczy średnią, odchylenie itp. czasów połączeń

## 3. billing

struct Polaczenie - przechowuje dane jednego połączenia  
class Billing - wczytuje i przechowuje listę połączeń,  
drukuje statystyki

# main.cpp



```
#include "billing.hpp"
```

```
int main ()  
{
```

```
    Billing bil (std::cin);
```

```
    std::cout << "*** STATYSTYKA DZIENNA ***" << std::endl;  
    bil.statystykaDzienna (std::cout);
```

```
    std::cout << "\n*** STATYSTYKA KRAJOWA ***" << std::endl;  
    bil.statystykaKrajowa (std::cout);
```

```
    return 0;
```

```
}
```

# histogram.hpp



```
#ifndef _histogram_hpp_
#define _histogram_hpp_

#include <vector>
#include <iostream>
#include <string>

class Histogram
{
    public:
        Histogram ();
        void dodaj (float x);
        size_t rozmiar () const;
        float srednia () const;           // użyj std::for_each() !
        float odchylenie () const;       // użyj std::for_each() !
        float max () const;              // użyj std::max_element() !
        float min () const;              // użyj std::min_element() !

    private:
        std::vector<float> dane_;
        mutable bool srednia_akt_;
        mutable bool odchylenie_akt_;
        mutable float srednia_;
        mutable float odchylenie_;
};

#endif
```

# billing.hpp



```
#ifndef _billing_hpp_
#define _billing_hpp_

#include <iostream>
#include <string>
#include <algorithm>
#include <vector>

struct Polaczenie
{
    unsigned dzien;
    std::string nr;
    float czas;           // czas połączenia
    std::string kod () const; // zwraca dwucyfrowy kod międzynarodowy
};

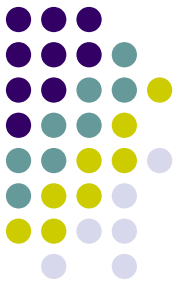
std::istream &operator>> (std::istream &is, Polaczenie &p);

class Billing
{
public:
    explicit Billing (std::istream &is);           // wczytuje połączenia z is
    void statystykaDzienna (std::ostream &os) const;
    void statystykaKrajowa (std::ostream &os) const;

private:
    std::vector <Polaczenie> blng_; // tablica połączeń
};

#endif
```

# billing.cpp



```
#include "billing.hpp"
#include "histogram.hpp"
#include <map>

// wybrane fragmenty

void Billing::statystykaDzienna (std::ostream &os) const
{
    std::vector <unsigned> stat (32, 0); // przechowuje liczbę połączeń każdego dnia
    // Przelatuje całą tablicę blng_ i uzupełnia tablicę stat
    // Formatuje i wyświetla wyniki na os
}

void Billing::statystykaKrajowa (std::ostream &os) const
{
    std::map <std::string, Histogram> stat; // osobna statystyka dla każdego kodu
    // Przelatuje całą tablicę blng_ i wrzuca czasy rozmów do
    // odpowiednich histogramów w stat

    // Formatuje i wyświetla wyniki na os
}
```

# Programowanie jest fantastyczne!!!

