

Środowisko pracy

Jako że lądujemy w tym semestrze w innym laboratorium, pierwszym zadaniem będzie przygotowanie środowiska pracy. Pozostaniemy przy tej samej wersji Code::Blocks, bo po pierwsze nowszej wersji nie ma, a po drugie nie trzeba będzie dla zachowania kompatybilności zmieniać instalacji domowej.

Jako że jesteście Państwo otrząskani w bojach, to tę część ćwiczenia zakończycie pokazaniem – oczywiście na sprzęcie laboratoryjnym – domyślnego programu C++ dostarczanego przez Code::Blocks zatrzymanego pułapką na instrukcji wyprowadzania napisu na konsolę.

Dojście do tego punktu (miejmy nadzieję, że sprawne) będzie warte dwa punkty, bo przygotowanie środowiska to pierwsze małe zadanie.

Klasa range

Przypuszczenie, że na przygotowanie środowiska nie pójdzie za dużo czasu, wydaje się uzasadnione.

Powinno nam zatem starczyć czasu na przypomnienie sobie, jak implementuje się klasę. W szczególności klasę range, która ma reprezentować przedział obustronnie otwarty.

Zestaw operacji nie jest specjalnie rozbudowany:

- konstruktor domyślny i z granicami przedziału,
- metody `low()` i `high()` do pobierania granic przedziału,
- metody `left_paren()` i `right_paren()` do pobierania znakowej reprezentacji lewej i prawej granicy przedziału,
- metoda `inrange()`, która sprawdza, czy argument jest we wnętrzu przedziału.

Klasa range

Wprowadzimy jeden element powielający wymienioną już funkcjonalność, mianowicie operator ().

Jego znaczenie będzie takie jak metody `inrange()`, a będziemy mogli dzięki niemu używać obiektów klasy `range` tak, jakby były funkcjami:

```
range test35(3, 5);  
int test_value = 4;  
if (not test35(test_value))  
    std::cout << "Error: " << test_value  
        << " not in range " << test35 << '\n';
```

Z tego krótkiego przykładu wynika także, że do zaimplementowania jest operator `<<` (tu powinien wyświetlić `(3, 5)`)