Sprawozdanie 13.04.2024

Sebastian Tarczyński niest. 1 rok, LK4

Przeciążenia operatorów

Zadanie 1. Zdefiniowana klasa Set (Zbiór), która reprezentuje zbiory elementów liczb całkowitych. Wszystkie zbiory mogą liczyć maksymalnie 10 elementów. Do przechowywania elementów zbioru użyty jest wektor typu całkowitego. Zdefiniowane zostały metody:
- wyświetlania elementów zbioru
- obliczania iloczynu zbiorów
- obliczania sumy zbiorów
- obliczania realizacji zawierania się zbiorów
- dodawania pojedynczego elementu do zbioru

Powyższe metody mają przypisane operatory, które będą realizowały operacje związane daną metodą

Kod:

Screen z terminala po wywołaniu

```
dev@dev-mbp-1 13-04 % cd "/Users/dev/Documents/projects/sem2/ppc/13-04/" && g++ -std=c++11 zad.cpp -o zad && "/Users/dev/Documents/projects/sem2/ppc/13-04/"zad
Sum: { 1 2 2 3 }
Intersection: { 2 }
A after increment: { 2 3 }
dev@dev-mbp-1 13-04 %
```

Zadanie 2.

Zdefiniowana klasa o nazwie CustomString, która służy do przechowywania łańcuchów tekstowych. Maksymalna długość łańcucha testowego wynosi 100 znaków. Zawiera metodą dodawania dwóch łańcuchów tekstowych z przypisanym operatorem

Kod:

Untitled (Workspace) - zad2.cpp

```cpp
#include <iostream>

using namespace std;

class CustomString
{
private:
    char text[100];

public:
    CustomString(const char *s = "")
    {
        int length = 0;
        while (s[length] != '\0')
        {
            ++length;
        }
        for (int i = 0; i < length; ++i)
        {
            text[i] = s[i];
        }
        text[length] = '\0';
    }

    CustomString operator+(const CustomString &other) const
    {
        CustomString result;
        int i = 0;
        while (text[i] != '\0')
        {
            result.text[i] = text[i];
            ++i;
        }
        int j = 0;
        while (other.text[j] != '\0')
        {
            result.text[i + j] = other.text[j];
            ++j;
        }
        result.text[i + j] = '\0';
        return result;
    }

    void print() const
    {
        cout << text << endl;
    }
};

int main()
{
    CustomString A("ALFA");
    CustomString B("BET");

    CustomString C = A + B;

    cout << "A: ";
    A.print();
    cout << "B: ";
    B.print();
    cout << "C: ";
    C.print();

    return 0;
}
```

Screen z terminala: