

Zadanie 8

Proszę przeczytać rozdziały na temat klas (w szczególności przeciążania operatorów) oraz przejrzeć rozdział na temat napisów.

Następnie, nie używając klasy `string` z nagłówka `<string>`, zdefiniować własną klasę `String`, której obiekty przechowują napisy (wskaźniki do tablic znaków) o dowolnej długości.

Można używać funkcji w rodzaju `strlen` i `strcpy` z nagłówka `<cstring>` oraz funkcji operujących na znakach, jak `tolower` czy `toupper`, z nagłówka `<cctype>` — patrz rozdział 17 wykładów.

Zdefiniować odpowiednie konstruktory i destruktor. Zdefiniować też metody tak, że po `String s("jakis NAPIS");`

- `s.toUpperCase()`; zmienia napis na "JAKIS NAPIS" i zwraca odniesienie (referencję) do `s` po zmianie
- `s.toLowerCase()`; zmienia napis na "jakis napis" i zwraca odniesienie (referencję) do `s` po zmianie
- `s.length()`; zwraca długość napisu (nie licząc ewentualnego znaku pustego kończącego napis)

Przeciążyć też operatory

- `'<<'` tak, aby możliwe było wyprowadzanie przechowywanych napisów na ekran (lub do dowolnego strumienia wyjściowego)
- `'=='` i `'!='` tak, aby `s1==s2` działało tak, jak w Javie `s1.equals(s2)` oraz `!s1.equals(s2)`. Powinno działać też porównywanie ze zwykłymi C-napisami (np. `s != "literal napisowy"`)
- przypisania `'='`
- dodawania napisów (jak w Javie): `s="abc"+s1+s2+"def"`

Na przykład poniższy program

```
#include <iostream>
#include <cstring>
#include <cctype>
using namespace std;

class String {
    char* nap;
public:
    String(const char* n);
    String(const String& s);
```

```

String operator+(const String& s) const;
String operator+(const char* n) const;
String& operator=(const String& s);
bool operator==(const String& s) const;
bool operator!=(const String& s) const;
String& toLower();
String& toUpper();
size_t length() const;

~String();
friend String operator+(const char*,const String&);
friend ostream& operator<<(ostream&, const String&);
};

int main(void) {
String s = "ALA " + String("ma ") + "kota";
cout << s << endl;
if (s == "ALA ma kota")
    cout << s.toUpper() << endl;
s = s = "KONIEC PROGRAMU"; // <- sic!
cout << s.length() << " " << s << endl;
}

```

po uzupełnieniu implementacji powinien się skompilować i wypisać

```

ALA ma kota
ALA MA KOTA
15 KONIEC PROGRAMU

```

Rozwiązania, w postaci **jednego** pliku źródłowego zawierającego treść programu, proszę wrzucać w systemie EDU do katalogu „Foldery zadań / Zadanie_08”

Nazwą pliku powinno być nazwisko z dużej litery (bez polskich znaków); rozszerzeniem musi być '.cpp', czyli np. *Malinowska.cpp*.