Stringi

- do tej pory posługiwaliśmy się tak zwanymi stringami typu `C`. Przypomnijmy, taki string jest tablicą znaków, zakończoną znakiem `NULL`. Deklaruje się je jako tablice, i sama nazwa stringu jest wskaźnikiem typu `char`. W szczególności intuicyjne podstawienie jednego stringu za drugi nie działa, wymagana jest funkcja `strcpy()` z biblioteki `string.h` lub `cstring`
- `C++` wprowadza klasę `string`, zdefiniowaną w bibliotece `string`, która „opakowuje" stringi typu `C`
- klasa `string` oprócz samego stringu oferuje nam różne funkcje ułatwiające posługiwanie się stringiem.
- na przykład, nie musimy martwić się rozmiarem stringu. Obiekt sam przydziela odpowiednią pamięć, także jeżeli nadpisujemy, wydłużamy string
- Przykład `prog1.cpp`

Stringi

- wiele możliwości inicjalizacji stringów i działania na nich
- Przykład `prog2.cpp`
- prosta zmiana długości stringu, program sam się tym martwi
- Przykład `prog3.cpp`
- stringi możemy przeszukiwać
- Przykład `prog4.cpp`

Strumienie stringowe

- możemy komunikować się ze stringiem przy pomocy schematu strumienia
- biblioteka `strstream` definiuje strumienie do komunikowania się z tablicami znaków bezpośrednio w pamięci - czyli C-stringami, zakończonymi znakiem `'\0'`

```
istrstream::istrstream( char *buf);
istrstream::istrstream( char *buf, int size)
```

- pierwszy konstruktor oczekuje wskaźnika do tablicy znaków zakończonej znakiem `'\0'`
- tablicę trzeba wcześniej samemu utworzyć
- drugi konstruktor nie potrzebuje znaku `'\0'` w tablicy - można ją odczytywać do `buf[size]`
- Przykład: `prog5.cpp`

Strumienie stringowe

- strumień wyjściowy korzysta z konstruktora:

```
ostrstream::ostrstream( char *,
                        int, int = ios::out);
```

- pierwszy parametr to wskaźnik do istniejącej tablicy znaków (trzeba zarezerwować pamięć wcześniej)
- drugi parametr to rozmiar tablicy
- trzeci parametr to tryb dostępu. Domyślnie zapisywanie zacznie się na początku bufora
- jeżeli ustalimy tryb na `ios::ate` lub `ios::app` to strumień wyszuka znak `'\0'` w buforze (koniec istniejącego stringu) i od tego miejsca dopisze
- strumień nie dopisuje automatycznie znaku `'\0'`. Trzeba to zrobić „ręcznie" aby zakończyć string. Służy do tego manipulator `ends`
- Przykład: `prog6.cpp`

Strumienie stringowe

- w przypadku strumienia wyjściowego `ostrstream` pamięć może być przydzielona automatycznie. Używamy wtedy konstruktora bez parametrów

```
ostrstream napis;
```

- tak utworzony strumień sam zajmuje się przydzielaniem pamięci w miarę potrzeb (kiedy string zmienia długość). Funkcja `napis.str()` zwraca wskaźnik na początek właściwego stringu. Ten wskaźnik może się zmieniać
- Przykład: `prog7.cpp`
- nowa biblioteka `sstream` służy do komunikowania się za stringami C++.
- strumienie `istringstream` i `ostringstream` same zajmują sie alokacją i zwalnianiem pamięci. Posługujemy się nimi podobnie jak wszystkimi strumieniami
- Przykład: `prog8.cpp`