

**Łańcuchy znaków**

Omówimy teraz kolejną grupę zmiennych, a mianowicie zmienne typu `string`, które służą do przechowywania tekstu lub ogólniej ciągu znaków. Sposób deklaracji takiej zmiennej wygląda tak samo jak każdej innej. Jedyną różnicą to sposób przypisania do niej konkretnej wartości (łańcucha znaków): tekst przypisany do zmiennej musi być objęty znakami " ". Oto przykład:

```
string napis = "Jan Kowalski";
```

Tekst możemy również podawać z klawiatury. Odbywa się to tak samo jak w przypadku zmiennych liczbowych (tutaj podajemy sam tekst bez znaków " " do momentu natrafienia na znak spacji, tabulacji bądź znaku nowego wiersza). Oto fragment kodu ilustrujący tę sytuację:

```
string imie;  
cout<< "Podaj swoje imie";  
cin>>imie;
```

Jeśli chcemy podać od razu np. imię i nazwisko (ogólnie chodzi o tekst oddzielony „białym znakiem” czyli m.in. spacją, tabulacją bądź znakiem nowego wiersza) musimy użyć składni z poleceniem `getline` w następujący sposób:

```
string dane;  
cout<<"Podaj swoje imie i nazwisko ";  
getline(cin, dane);
```

Warto zwrócić uwagę na to, że każda zmienna typu `string` jest tak naprawdę tablicą, której elementami są poszczególne znaki występujące w łańcuchu przypisanemu do tej zmiennej. Przykładowo zmienna `napis` jest tablicą składającą się z 12 elementów (spacja jest traktowana jako jeden znak), a wyraz `napis[5]` to litera `o` (indeksowanie tablicy zaczyna się od zera).

Na łańcuchach znaków możemy wykonywać szereg operacji. Większość z nich realizowana jest za pomocą pewnych funkcji (metod), które znajdują się w bibliotece o nazwie `string`. Stąd też, gdy piszemy program, w którym wykorzystujemy te funkcje należy dodać w odpowiednim miejscu wiersz

```
#include<string>
```

Jedną z podstawowych operacji jest dodawanie (konkatenacji) ciągów znaków. Odbywa się to za pomocą operatora `+`. Poniżej przedstawiamy fragment kodu, który wyświetli napis: `Jan Kowalski`

```
string imie = "Jan";  
string nazwisko = "Kowalski";  
string dane = imie + " " + nazwisko;  
cout<<dane<<endl;
```

Ten sam efekt możemy uzyskać stosując metodę `append()` następująco:

```
string imie = "Jan";  
string dane = imie.append(" Kowalski");  
cout<<dane<<endl;
```

Zauważmy, że do wywołania metody `append()` używamy tzw. notacji z kropką:

```
nazwa_zmiennej.metoda(argumenty).
```

W ten sam sposób będziemy wywoływać pozostałe omawiane funkcje.

Kolejną z nich to `substr()`. Jest to dwuargumentowa funkcja, która służy do generowania podciągów, a dokładniej zwraca część danego ciągu. Pierwszym argumentem jest pozycja początku wygenerowanego podciągu, a drugi to liczba znaków w tym podciągu. Przykładowo poniższy fragment kodu spowoduje wyświetlenie tekstu: `kot`

```
string napis = "maskotka";  
string napis_1 = napis.substr(3,3);
```

```
cout<<napis_1;
```

Następna funkcja to `size()`. Jest to bezargumentowa funkcja, która zwraca liczbę znaków występujących w danym łańcuchu znaków. Przykład

```
string napis = "bolek";  
int s = napis.size();  
cout<<s; // wyświetli 5
```

Funkcja `insert()` z kolei służy do wstawiania do danego ciągu znaków innego ciągu znaków. Pierwszym argumentem tej funkcji jest pozycja początku nowego ciągu znaków, zaś drugim – wstawiany ciąg. Oczywiście funkcja zwraca powstały w ten sposób nowy ciąg znaków. Oto przykład:

```
string napis_1 = "W puszczy";  
string napis_2 = napis_1.insert(1, " pustyni i w");  
cout<<napis_2; //wyświetli W pustyni i w puszczy
```

Jeśli chcemy zastąpić część danego ciągu przez inny ciąg możemy użyć funkcji `replace()`. Jest to funkcja trójargumentowa: pierwszym argumentem jest pozycja początku ciągu, którym chcemy zastąpić część danego ciągu, drugi argument to ilość znaków do zastąpienia, a trzeci to ciąg zastępczy. Oto przykład:

```
string ciag = "Jan Kowalski";  
string ciag_1 = ciag.replace(6,3, "chanow");  
cout<<ciag_1; //wyświetli Jan Kochanowski
```

Następna funkcja o nazwie `toupper()` pozwoli zmienić małe litery na wielkie. Kod

```
string znak = toupper('a');  
cout<<znak;
```

wyświetli literę A. Aby użyć tej funkcji należy dołączyć bibliotekę `cctype`.

Ostatnią funkcją, którą omówimy jest funkcja `find()`. Zwraca ona miejsce (indeks tablicy) pierwszego wystąpienia w ciągu zadanego podciągu lub znaku. Przykład:

```
string napis = "lokomotywa";  
int s = napis.find("mot");  
cout<<s; //wyswietli 4
```

### Zadanie 1.

Napisać program (w pliku o nazwie `osoba.cpp`), który pobierałby od użytkownika podstawowe jego dane: imię, nazwisko, wiek, płeć (jako jeden znak M lub K), datę urodzenia (dd.mm.rrrr) oraz adres (osobno miejscowość, nazwa ulicy – musi być co najmniej dwuczłonowa np. ul. Jana Pawła II oraz nr/nr mieszkania np. 43/13) i wypisałby je na ekran w następujący sposób:

```
Nazywam sie imie nazwisko. Mam wiek lat i jestem M/K. Urodzilem sie data_ur, mieszkam w miejscowosc przy ulicy nazwa_ulicy nr/nr_mieszkania.
```

### Zadanie 2.

Napisać program testujący wszystkie omawiane wcześniej funkcje.

### Zadanie 3.

Napisać program, który wczyta nazwę dowolnego pliku z klawiatury (nazwa.rozszerzenie). Następnie program wypisze informację, czy dany plik jest plikiem źródłowym z c++ czy nie.