

**Laboratorium 4 Pomoc. Klasa string. Obiekty jako składowe klasy i parametry metod. Wskaźnik this. Statyczne składowe klasy. Tablice obiektów. Tablice wskaźników na obiekty.**

- **Obiekty klasy std::string (#include <string>)** – konstruktory

```
string napis1;           //string();
string napis2("Janek");  //string(const char *cstr);
char tablica[20]={"Kowalski"};
string napis3(tablica);
string napis4(&tablica[5]);
string napis5("Ala ma kota",8); //string(const char *cstr,
                                //size_type ile);
string napis6(25,'*'); // //string(size_type ile, char z);
string napis7="Kot ma Ale";
string napis8(napis7,4,2); //const string &st,
                            //size_type pozycja, size_type ile)
cout<<napis1<<"\n"<<napis2<<"\n"<<napis3<<"\n"
<<napis4<<"\n"<<napis5<<"\n"<<napis6<<"\n"<<napis7<<"\n"
<<napis8<<"\n";
```

- **Operatory =, +, += dla obiektów klasy string**

```
napis1=napis2; // kopiowanie
napis1=napis2+" "+napis3; //dodawanie
napis1+=napis7; //dołączanie
```

- **Wprowadzanie danych do C-stringów i stringów**

```
char napis[100];
cin>>napis; //wczytuje jedno słowo
cin.getline(napis,100); //wczytuje linię, pomija \n
cin.get(napis, 100); //wczytuje linię, ustawia \n w kolejce

string napis;
cin>>napis; //wczytuje jedno słowo
getline(cin,napis); //wczytuje linię, pomija \n
```

- **Wybrane metody klasy string**

- **size(), length()** – rozmiar i długość: liczba znaków (ta sama wartość)
- **max\_size()** – maksymalna długość tekstu dla obiektu
- **empty()** – sprawdza czy string jest pusty
- **clear()** – wyczyszczenie -> pusty string
- **at()** – znak na pozycji (lub operator[] ) (cout<<napis2.at(3)<< napis2[3];)
- **substr(pozycja, ile)** – fragment stringu (substring)
- **find(substring, odkądSzukać)** – szukanie od początku substringu w obiekcie klasy string (zwraca pozycję początku substringu – gdy znaleziono lub string::npos (największa możliwa w danym komputerze liczba całkowita, nieskończoność) – gdy nie znaleziono
- **erase(skąd, ileZnaków)** – usuwanie znaków ze stringu
- **insert(gdzieWstawić, coWstawić)** – wstawianie znaków do stringu
- **replace(odkąd, ileZastąpić, coWstawić)** – zamiana części znaków na inne

- **Tożsamość obiektu - wskaźnik *this*** – wskazuje na obiekt, dla którego została wywołana metoda lub odwołanie do pola obiektu (odróżnienie od pól struktury, zmiennych lokalnych lub parametrów o tych samych nazwach), do wykorzystania w metodach klasy

```
Tdata::Tdata(int d, int m, int r)
{ this->d=d; this->m=m; this->r=r; }
```

- **Statyczne składowe klasy** (są przypisane do klasy, a nie jej instancji - obiektów):
  - pola (dzielone przez wszystkie obiekty do komunikacji między obiektami)  
**class nazwaKlasy {...static typ nazwa;...};**  
Konieczna definicja poza klasą z ew. inicjalizacją:  
**int nazwaKlasy::nazwaPola [=wartośćPoczątkowa];**
  - metody (niezależne od obiektów, nie mogą wykorzystywać wskaźnika **this**)  
**class nazwaKlasy {...static typ nazwa(); ...};**
- **Odwołanie do składnika statycznego:**
  - **Klasa::SkładnikStatyczny**
  - **Obiekt.SkładnikStatyczny** - jeżeli istnieją już jakieś obiekty tej klasy
  - **Wskaźnik-> SkładnikStatyczny** – jeżeli został zdefiniowany wskaźnik do klasy:  
**Klasa \* Wskaźnik**

```
const int DL=20;
class TosobaS
{ private:
    static int liczbaObiektow;//pole statyczne
protected:
    char nazwisko[DL], imie[DL];
    int wiek;
public:
    TosobaS();
    TosobaS(char *nazwisko, char *imie, int wiek);
    void podajDane();
    void wyswietl();
    ~TosobaS();
    static int ile();//metoda statyczna
};
int TosobaS::liczbaObiektow;
```

- **Tablica obiektów**

```
const int n=2;
Tosoba osoba3[n]; // tablica obiektów - domyślny konstruktor
for(int i=0; i<n; i++)
    osoba3[i].podajDane();
```

Konstruktor jest wywoływany dla każdego elementu tablicy - klasa musi mieć konstruktor domyślny -

- **Tablica wskaźników na obiekty (new, delete)**

```
Tosoba *wskosoba4[3]; // tablica wskaźników na obiekty
for(int i=0; i<n; i++)
{
    wskosoba4[i]=new Tosoba; // Utworzenie obiektu dynamicz.
    wskosoba4[i]->podajDane();
}
```

- **Wskaźnik do tablicy obiektów (new [], delete[])**

```
Tosoba *wskosoba5;
wskosoba5=new Tosoba[n];
for(int i=0; i<n; i++)
{
    (wskosoba5+i)->podajDane();
}
delete [] wskosoba5;
```