

Laboratorium 3 Pomoc. Metody klasy. Konstruktory, destruktory. Przetwarzanie obiektów przez zmienną, wskaźnik i referencję

Cel laboratorium:

Definiowanie klas z różnymi metodami. Zapoznanie z przeciążaniem metod, w tym z przeciążaniem konstruktorów oraz z różnymi sposobami przetwarzania obiektów: przez zmienną, wskaźnik, referencję.

- **Metody klasy:**
 - Funkcje zwykłe (z parametrami lub bez)
class nazwaKlasy {...typ nazwa(parametry);...};
 - Funkcje stałe (nie zmieniają pól)
class nazwaKlasy {... typ nazwa() const; ...};
 - **Konstruktor** – publiczna funkcja wywoływana automatycznie podczas tworzenia obiektu: inicjowanie pól, przydzielanie pamięci – może być kilka
nazwaKlasy(); //bez typu
nazwaKlasy(parametry);
 - **Destruktor** – publiczna funkcja wywoływana zawsze podczas niszczenia obiektu – tylko jeden
~ nazwaKlasy(); //bez typu
- **Przeciążanie metod** -definiowanie w klasie kilku funkcji (często konstruktorów) o tej samej nazwie a różnych parametrach; sposób wywołania określa która to funkcja
- **Przesłanianie nazw** – identyczna nazwa zmiennych, pól w klasie, zmiennych lokalnych metod

nazwaKlasy::nazwa; //nazwa pola w klasie
- **Rodzaje konstruktorów:**
 - **Domyslny** –wywoływany podczas tworzenia obiektu, gdy nie podamy parametrów
 - **Z parametrami** - wywoływany podczas tworzenia obiektu, gdy podamy parametry
 - **Z listą inicjalizacyjną** – wywoływany podczas tworzenia obiektu z automatycznym nadaniem wartości polom za pomocą listy
 - **Kopiujący** – wywoływany podczas kopiowania obiektów
- Przykład

```
class Tdata
{ private:
    int d,m,r;
public:
    Tdata();           //K1 - domyslny
    Tdata(int dd, int mm); //K2 przeciazony
    Tdata(int dd, int mm, int rr):d(dd), m(mm), r(rr) {};
    //K3 z lista
```

```
Tdata (Tdata &); //K4 kopiujacy
~Tdata();
void wczytaj();
void drukuj();
int nastepnyRok() const;
};

Tdata::Tdata()
{ d=1; m=1; r=2015;
}
Tdata::Tdata(int dd, int mm) //K2 przeciazony
{ d=dd; m=mm; r=2015;
}
Tdata::Tdata (Tdata & ref)
{d=ref.d; m=ref.m; r=ref.r;
}

// class destructor
Tdata::~~Tdata()
{
    // insert your code here
}
void Tdata::wczytaj()
{cout<<"Podaj d "; cin>>d;
  cout<<"Podaj m "; cin>>m;
  cout<<"Podaj r "; cin>>r;
}
void Tdata::drukuj()
{cout<<d<<"."<<m<<"."<<r<<endl;}

int Tdata::nastepnyRok() const
{return r+1;}

//przetwarzanie obiektów
int main(int argc, char *argv[])
{
    Tdata data;
    data.wczytaj();
    data.drukuj();

    Tdata data1(15,10,2014);
    data1.drukuj();

    Tdata data2(21,7);
    data2.drukuj();

    Tdata data3=data;
    data3.drukuj();
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

- **Metody przetwarzania obiektów:**
 - Poprzez zmienną obiektową (.)
 - Poprzez wskaźnik na zmienną obiektową (->, new, delete)
 - Poprzez referencję (&)
- **Przetwarzanie obiektów przez zmienną**
 1. Utworzenie:
nazwaKlasy zmienna;
nazwaKlasy zmienna(parametry);
 2. Praca na obiekcie:
zmienna.pole zmienna.funkcja()
 3. Usunięcie obiektu: automatycznie przez środowisko lub poprzez destruktor
zmienna.~nazwaKlasy
- **Przetwarzanie obiektów przez wskaźnik**
 1. Utworzenie wskaźnika :
nazwaKlasy* wsk=new nazwaKlasy;
nazwaKlasy* wsk=new nazwaKlasy(parametry);
 2. Praca na wskaźniku: odwołanie do pola lub metody publicznej:
wsk->pole wsk->funkcja()
 3. Usunięcie wskaźnika
delete wsk;
- **Przetwarzanie obiektów przez referencję**
 1. Utworzenie referencji do obiektu
nazwaKlasy zmienna1;//utworzenie obiektu zmienna1
nazwaKlasy &zmienna2=zmienna1;//referencja do zmienna1
 2. Wykorzystanie referencji
zmienna2.wczytajPola();//wczytanie pól zmiennej1
zmienna2.wyswietlPola();//wyswietlenie pól zmiennej1

Inicjalizacja łańcuchów

```
class Tnowa
{ private:
char nazwa[30];
...
konstruktor:
Tnowa::Tnowa(const char *naz)
{ std::strncpy(nazwa, naz, 29); nazwa[29]='\0'; }

class Tnowa
{ private:
std::string nazwa;
...
konstruktor:
Tnowa::Tnowa(const char *naz)
// Tnowa::Tnowa(const std::string & naz)

{ nazwa=naz }
```