Laboratorium nr 13

Temat: Klasy, konstruktory, destruktory.

Zakres laboratorium:

- konstruktor
- destruktor
- konstruktor z argumentami domyślnymi
- konstruktor domniemany
- zadanie laboratoryjne

Konstruktor

- **Konstruktor** to specjalna funkcja składowa klasy, która ma nazwę identyczną z nazwą klasy.
- **Konstruktor** **NADAJE OBIEKTOWI KLASY WARTOŚĆ POCZĄTKOWĄ!!! (ściślej danym składowym obiektu)**.
- Przed nazwą **konstruktora** nie może być żadnego określenia typu zwracanego (ani `int`, ani `float`, ani nawet `void`).
- Jeśli klasa ma **odpowiedni konstruktor**, to jest on wywoływany automatycznie ilekroć powołujemy do życia nowy obiekt danej klasy.
- **Konstruktor** jest funkcją, przy której najczęściej spotyka się **przeciążenie nazwy**. Dzięki temu dane składowe klasy mogą być inicjalizowane w różny sposób.
- **Konstruktor** może wywołać jakąś inną funkcję składową swojej klasy.
- Jeśli w klasie nie ma zdefiniowanego żadnego konstruktora, kompilator wygeneruje tzw. **konstruktor domniemany**, jednakże nie wykona on żadnej inicjalizacji i dlatego po utworzeniu obiektu **nie ma gwarancji, że jego dane będą spójne!!!**.
- Kiedy to możliwe (czyli prawie zawsze) należy napisać funkcje konstruktora, aby zapewnić, że wszystkie dane składowe klasy zostaną zainicjalizowane we właściwy sposób. Szczególnie dane będące wskaźnikami powinny otrzymać odpowiednią wartość.

Destruktor

- **Destruktor** to specjalna funkcja składowa klasy, która ma nazwę identyczną z nazwą klasy poprzedzoną znaczkiem tyldy (~).
- **Destruktor** to jakby „sprzątaczka”, która sprząta tuż przed zlikwidowaniem obiektu, jednakże destruktor **NIE LIKWIDUJE OBIEKTU!!!**.
- Przed nazwą **destruktora** nie może być żadnego określenia typu zwracanego (ani `int`, ani `float`, ani nawet `void`).
- **Destruktor** wywoływany jest automatycznie ilekroć obiekt danej klasy ma być zlikwidowany.
- **Destruktor** jest wywoływany bez żadnych argumentów. W związku z tym jest funkcją, której **nazwy nie można przeciążyć**.
- **Destruktor** może wywołać jakąś inną funkcję składową swojej klasy.
- Jeśli w klasie nie ma zdefiniowanego jawnego destruktora, kompilator wygeneruje destruktor za ciebie.
- Kiedy to możliwe należy napisać funkcje destruktora, aby zapewnić wykonanie czynności końcowych przed zlikwidowaniem obiektu. Szczególnie dane będące wskaźnikami powinny w destruktorze zwalniać zajmowaną pamięć.

```
class Czas
{
   int godzina,minuta,sekunda;                    //prywatne dane składowe
public:
   Czas();                                        //konstruktor
   ~Czas();                                       //destruktor
   void ustaw_czas(int g,int m,int s);            //funkcja składowa klasy
   void drukuj_czas();                            //funkcja składowa klasy
};
////////////////////////////////////////////////////////////////////////
Czas::Czas()                                      //definicja konstruktora
{godzina=minuta=sekunda=0;}
////////////////////////////////////////////////////////////////////////
Czas::~Czas(){}                                   //definicja destruktora
////////////////////////////////////////////////////////////////////////
Czas::ustaw_czas(int g,int m,int s)
{godzina=g;minuta=m;sekunda=s;}
////////////////////////////////////////////////////////////////////////
Czas::drukuj_czas()
{cout<<godzina<<":"<<minuta<<":"<<sekunda<<endl;}
////////////////////////////////////////////////////////////////////////
main()
{
   Czas koniec_zajec;                             //definiujemy obiekt
   koniec_zajec.drukuj_czas();                    //0:0:0
   koniec_zajec.ustaw_czas(13,15,0);              //ustawiamy czas
   koniec_zajec.drukuj_czas();                    //13:15:0
}
```

Konstruktor z argumentami domyślnymi

- **Konstruktory** mogą posiadać wartości domyślne.
- Nie wszystkie argumenty w **konstruktorze z argumentami domyślnymi** muszą być domyślne.
- **Konstruktor domyślny** (ang. *default constructor*) to konstruktor dostarczany przez programistę, definiujący **dla wszystkich** swoich argumentów wartości domyślne.
- **Konstruktor domyślny** może być albo wywołany bez podania żadnych argumentów (wszystkie argumenty są wtedy domyślne), albo z ich podaniem.
- Domyślne argumenty w konstruktorze domyślnym deklaruje się wyłącznie w prototypie funkcji konstruktora w definicji klasy pliku nagłówkowego.
- Podanie w konstruktorze wartości domyślnych pozwala na poprawną inicjalizację składowych klasy nawet, jeżeli w jego wywołaniu nie zostały podane żadne wartości.
- W danej klasie może istnieć tylko jeden **konstruktor domyślny**.

```
class Czas
{
   int godzina,minuta,sekunda;                          //prywatne dane składowe
public:
   Czas(int=0,int=0,int=0);                             //konstruktor domyślny
   void ustaw_czas(int g,int m,int s);                  //funkcja składowa klasy
   void drukuj_czas();                                  //funkcja składowa klasy
};
////////////////////////////////////////////////////////////////////////
Czas::Czas(int g,int m,int s)                           //definicja konstruktora
{
   ustaw_czas(g,m,s);                                   //wywołanie funkcji
}
////////////////////////////////////////////////////////////////////////
Czas::ustaw_czas(int g,int m,int s)
{
   godzina=g;minuta=m;sekunda=s;
}
////////////////////////////////////////////////////////////////////////
main()
{
   Czas czas1;                                  //wszystkie argumenty domyślne
   Czas czas2(13);                              //minuty i sekundy domyślne
   Czas czas3(13,15);                           //sekundy domyślne
   Czas czas4(13,15,0);                         //podane wszystkie wartości
   Czas czas_bledny(31,76,89);                  //błędnie podane wartości
   //inny zapis - też poprawny, ale nieco dłuższy
   Czas czas1=Czas();                           //wszystkie argumenty domyślne
   Czas czas2=Czas(13);                         //minuty i sekundy domyślne
   Czas czas3=Czas(13,15);                      //sekundy domyślne
   Czas czas4=Czas(13,15,0);                    //podane wszystkie wartości
}
```

```
class Czas
{
public:
   Czas(int=0,int=0,int=0);          //konstruktor domyślny
};
////////////////////////////////////////////////////////////
```

Czyli tak, jakbyśmy mieli w klasie 4 następujące konstruktory:

```
class Czas
{
public:
   Czas(int,int,int);
   Czas(int,int);
   Czas(int);
   Czas();
};
```

Konstruktor domniemany

- **Konstruktor domniemany** to taki konstruktor, który jest wywoływany bez żadnych argumentów.

- Różnica pomiędzy **konstruktorem z argumentami domyślnymi** a **konstruktorem domniemanym** jest taka, że konstruktor z argumentami domyślnymi **może być** wywołany bez podania żadnych argumentów (**wtedy, gdy ma wszystkie argumenty domyślne**), natomiast konstruktor domniemany **zawsze jest** wywoływany bez podania żadnych argumentów.

- W świetle powyższej definicji konstruktorem, który można wywołać bez żadnych argumentów jest konstruktor ze wszystkimi argumentami domyślnymi, czyli **konstruktorem domniemanym jest jednocześnie konstruktor domyślny!!!**

- Klasa może mieć tylko jeden **konstruktor domniemany** (wynika to z istoty przeciążania funkcji)

```
class Czas
{
   int godzina,minuta,sekunda;                  //prywatne dane składowe
public:
   Czas(void);                                  //konstruktor domniemany
   void ustaw_czas(int g,int m,int s);          //funkcja składowa klasy
   void drukuj_czas();                          //funkcja składowa klasy
};
////////////////////////////////////////////////////////////////////////
Czas::Czas(void)                                //definicja konstruktora
{
   drukuj_czas();                               //wywołanie funkcji
}
////////////////////////////////////////////////////////////////////////
Czas::ustaw_czas(int g,int m,int s)
{
   godzina=g;minuta=m;sekunda=s;
}
////////////////////////////////////////////////////////////////////////
main()
{
   Czas czas1;                          //wywołanie bez argumentów!!!
   //lub
   Czas czas1=Czas();                   //wywołanie bez argumentów!!!
}
```

```
class Czas
{
public:
   Czas(char,int,int=0);    //konstruktor z 1 argumentem domyślnym
   Czas(int=0,int=0,int=0);    //konstruktor domyślny (domniemany)
   //Czas(void);               //konstruktor domniemany
};
////////////////////////////////////////////////////////////
```

Zadanie laboratoryjne