# Programowanie w C++ - klasy

Anna Gogolińska

# Klasy

- O klasie możemy myśleć jak o własnym typie, w którym możemy mieć:
  - Wartości – nazywane polami klasy.
  - Funkcje – nazywane metodami. One często służą do zmiany wartości pól.
- Klasy określają pewien rodzaj bytów, np. klasa samochód, klasa punkt, klasa trójkąt.
  - Na przykład dla klasy samochód polami mogą być rok produkcji, przebieg, pojemność silnika, właściciel. Metody to mogą być zmiana przebiegu lub zmiana właściciela.

# Obiekty

- Jeśli klasa to typ, to zmienne tego typu nazywamy obiektami.
- Obiekty są realizacją klasy, zawierają te elementy, które są określone w ich klasie (pola i metody) tylko już z konkretnymi wartościami.
  - Na przykład dla klas samochód obiekt to będzie samochód z ustalonym rokiem produkcji, przebiegiem, właścicielem itd.

# Konstruktory/destruktor

- Konstruktor to specjalna metoda, która jest używana gdy tworzymy obiekt danej klasy.
  - Może mieć argumenty lub nie.
  - Może być kilka konstruktorów – muszą różnić się argumentami.
  - Zazwyczaj w konstruktorach ustawia się początkowe wartości dla pól.
- Destruktor to metoda wykonywana gdy obiekt jest niszczony – powinien zawierać czyszczenie pamięci itd..

# Modyfikatory dostępu

- Elementy klasa mogą mieć różną „widoczność”, dostęp. Ma to znaczenie mając obiekt klasy chcemy użyć jego elementów.
  - **public** – elementy są dostępne poza obiektem, można ich używać w kodzie używając nazwy obiektu.
  - **private** – do elementów jest dostęp tylko wewnątrz obiektu, czyli w kodzie klasy.
  - **protected** – jak private ale elementy są dostępne w klasach potomnych.

# Przykład

```
class Osoba {
    public:
        int wiek;
        string imie;
        string nazwisko;
        Osoba() {
                imie = "";
                nazwisko = "";
                …
        }
        Osoba(string i, string n, int w, int p) {
                imie = i;
                nazwisko = n;
                ustawPesel(p);
                …
        }
        ~Osoba() { …. }
        void ustawWiek(int w) {
                wiek = w;
        }

    private:
        int pesel;
        void ustawPesel(int p) {
                pesel = p;
        }
```

**class** znaczy że definiujemy klasę.

```
class Osoba {
    public:
        int wiek;
        string imie;
        string nazwisko;
        Osoba() {
                imie = "";
                nazwisko = "";
                …
        }
        Osoba(string i, string n, int w, int p) {
                imie = i;
                nazwisko = n;
                ustawPesel(p);
                …
        }
        ~Osoba() { …. }
        void ustawWiek(int w) {
                wiek = w;
        }

    private:
        int pesel;
        void ustawPesel(int p) {
                pesel = p;
        }
```

Piszemy modyfikator dostępu (private, public), a potem wymieniamy elementy o tym modyfikatorze.

```
class Osoba {
    public:
        int wiek;
        string im
        string na
        Osoba() {
                imie = "";
                nazwisko = "";
                …
        }
        Osoba(string i, string n, int w, int p) {
                imie = i;
                nazwisko = n;
                ustawPesel(p);
                …
        }
        ~Osoba() { …. }
        void ustawWiek(int w) {
                wiek = w;
        }

    private:
        int pesel;
        void ustawPesel(int p) {
                pesel = p;
        }
```

# Przykład

```
class Osoba {
    public:
        int wiek;
        string imie;
        string nazwisko;
        Osoba() {
            imie = "";
            nazwisko = "";
            …
        }
        Osoba(string i, string n, int w, int p) {
            imie = i;
            nazwisko = n;
            ustawPesel(p);
            …
        }
        ~Osoba() { …. }
        void ustawWiek(int w) {
            wiek = w;
        }

    private:
        int pesel;
        void ustawPesel(int p) {
            pesel = p;
```

Konstruktor: brak typu zwracanego i nazwa jak nazwa klasy.

Destruktor: brak typu zwracanego, tylda i nazwa klasy.

# Przykład

```
int main() {
    Osoba o1;
    Osoba o2("Jan", "Kowal", 34, 1234567);
    cout << o2.imie << " " << o2.nazwisko << " " << o2.wiek << endl;
    o2.ustawWiek(35);
    cout << o2.wiek << endl;

    return 0;
}
```

# Dziedziczenie

- Jednak klasa może dziedziczyć, czyli rozszerzać inną.
  - Klasa dziedzicząca – klasa potomna. Klasa z której się dziedziczy – klasa nadrzędna.
  - Klasa potomna ma te same elementy co klasa nadrzędna (poza elementami private).
  - Klasa potomna może zawierać nowe elementy.

# Przykład

```
class Pracownik : public Osoba {
//elementy private w Osoba należy zmienić na protected
    public:
        int id;
        Pracownik() { … }
        Pracownik(….) { …}
        //klasa ta ma również elementy z Osoba
}
```