

DZIEDZICZENIE KLAS

1. Co to jest dziedziczenie?

Dziedziczenie (ang. *inheritance*) – mechanizm współdzielenia funkcjonalności między klasami. Klasa może dziedziczyć po innej klasie, co oznacza, że oprócz swoich własnych atrybutów oraz metod, uzyskuje także te pochodzące z klasy, z której dziedziczy. Klasa dziedzicząca jest nazywana **klasą pochodną** lub **potomną** (**subclass** lub **derived class**), zaś klasa, z której następuje dziedziczenie — **klasą bazową** (**superclass**). Z jednej klasy bazowej można uzyskać dowolną liczbę klas pochodnych. Klasy pochodne posiadają obok swoich własnych metod i pól, również kompletny interfejs klasy bazowej.

Klasa potomna może na nowo implementować odziedziczone metody oraz dodawać nowe funkcje i pola, może też pozostać przy odziedziczonej zawartości. Dzięki dziedziczeniu unikamy więc wielokrotnego powtarzania fragmentów kodu. Oprócz tego hierarchie klas pomagają w utrzymaniu przejrzystości struktury programu.

2. Modyfikatory dostępu

W programowaniu obiektowym rozróżniamy 3 modyfikatory dostępu:

- **public** - atrybut lub metoda jest widoczna zarówno ze środka klasy, klas dziedziczonych jak i z zewnątrz klasy
- **protected** - atrybut lub metoda nie jest widoczna na zewnątrz klasy i może być dziedziczona
- **private** - atrybut lub metoda nie jest widoczna na zewnątrz klasy i nie może być dziedziczona.

Składowe prywatne mogą być jednak dostępne dla klas pochodnych – np. za pomocą odziedziczonych nieprywatnych metod z klasy bazowej.

3. Specyfikatory widoczności (C++)

```
class nazwa_klasy : [operator_widoczności] nazwa_klasy_bazowej
{
    // definicja klasy
};
```

```
class KlasaPochodna: public KlasaBazowa
```

Operator widoczności może przyjmować jedną z trzech wartości: **public**, **protected**, **private**. Operator widoczności przy klasie, z której dziedziczymy pozwala ograniczyć widoczność elementów publicznych z **klasy bazowej**.

public - oznacza, że dziedziczone elementy (np. zmienne lub funkcje) mają taką widoczność jak w klasie bazowej, czyli:

public → *public*

protected → *protected*

private → brak dostępu w klasie pochodnej

protected - oznacza, że elementy publiczne zmieniają się w chronione.

public → *protected*

protected → *protected*

private → brak dostępu w klasie pochodnej

private - oznacza, że wszystkie elementy klasy bazowej zmieniają się w prywatne.

public → *private*

protected → *private*

private → brak dostępu w klasie pochodnej

brak operatora - oznacza, że niejawnie (domyślnie) zostanie wybrany operator **private**

4. Dziedziczenie wielokrotne

Niektóre języki programowania (C++, Perl) obsługują tzw. dziedziczenie wielokrotne. W standardowym dziedziczeniu (pojedynczym) klasa pochodna dziedziczy po dokładnie jednej klasie bazowej. W dziedziczeniu wielokrotnym klas bazowych może być więcej:

```
class Klasa: public BazowaA, public BazowaB {  
public:  
    void wyzeruj() {  
        BazowaA::licznik = 0;  
        BazowaB::licznik = 0;  
    }  
};
```

Jeśli pola lub metody w powtarzają się, wówczas konieczna jest dodatkowa klasyfikacja nazwą klasy.

5. Przykładowe schematy dziedziczenia

