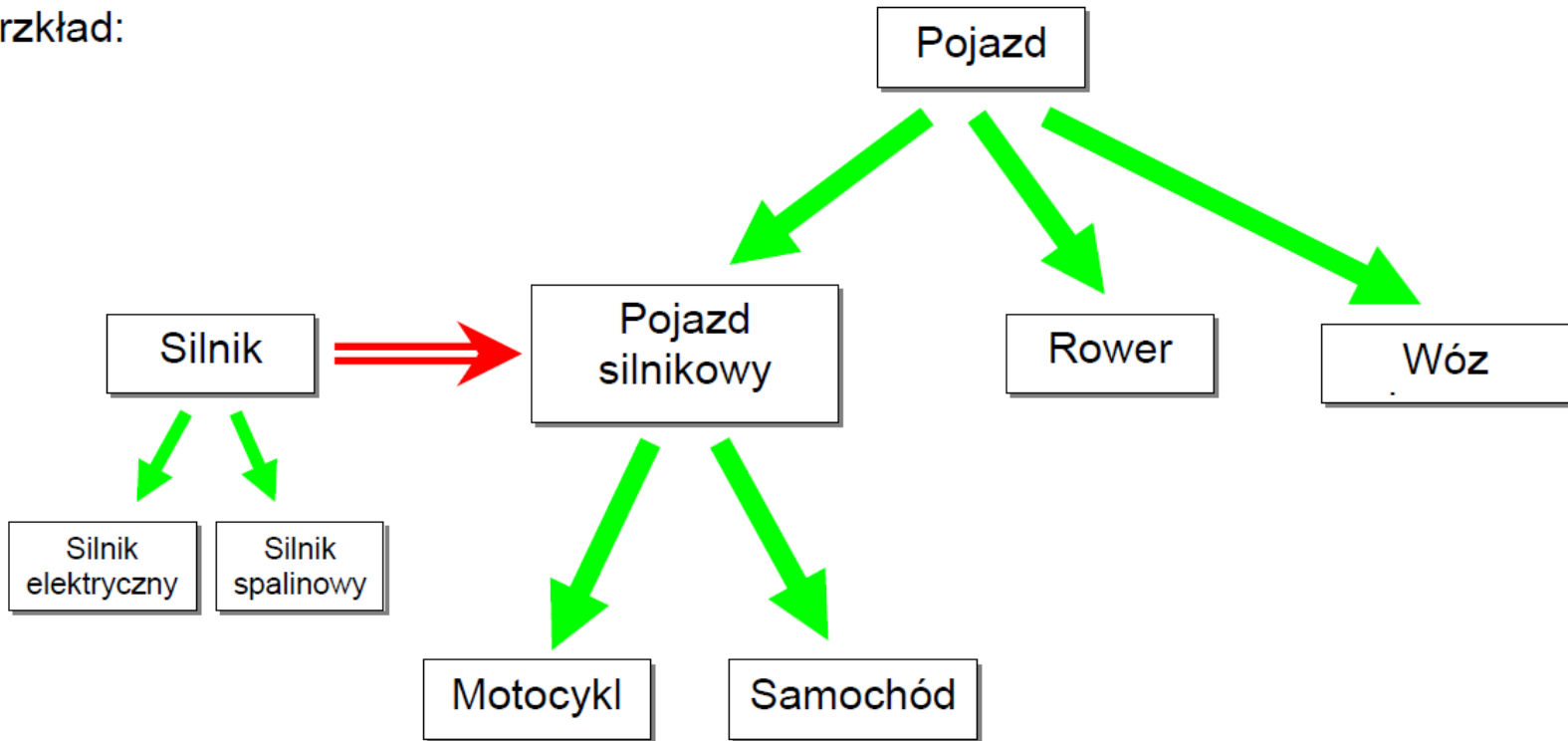


Programowanie obiektowe

09.05.2017

Związki między klasami: „jest” i „zawiera”

Przykład:



Pojazd silnikowy **jest** szczególnym rodzajem Pojazdu

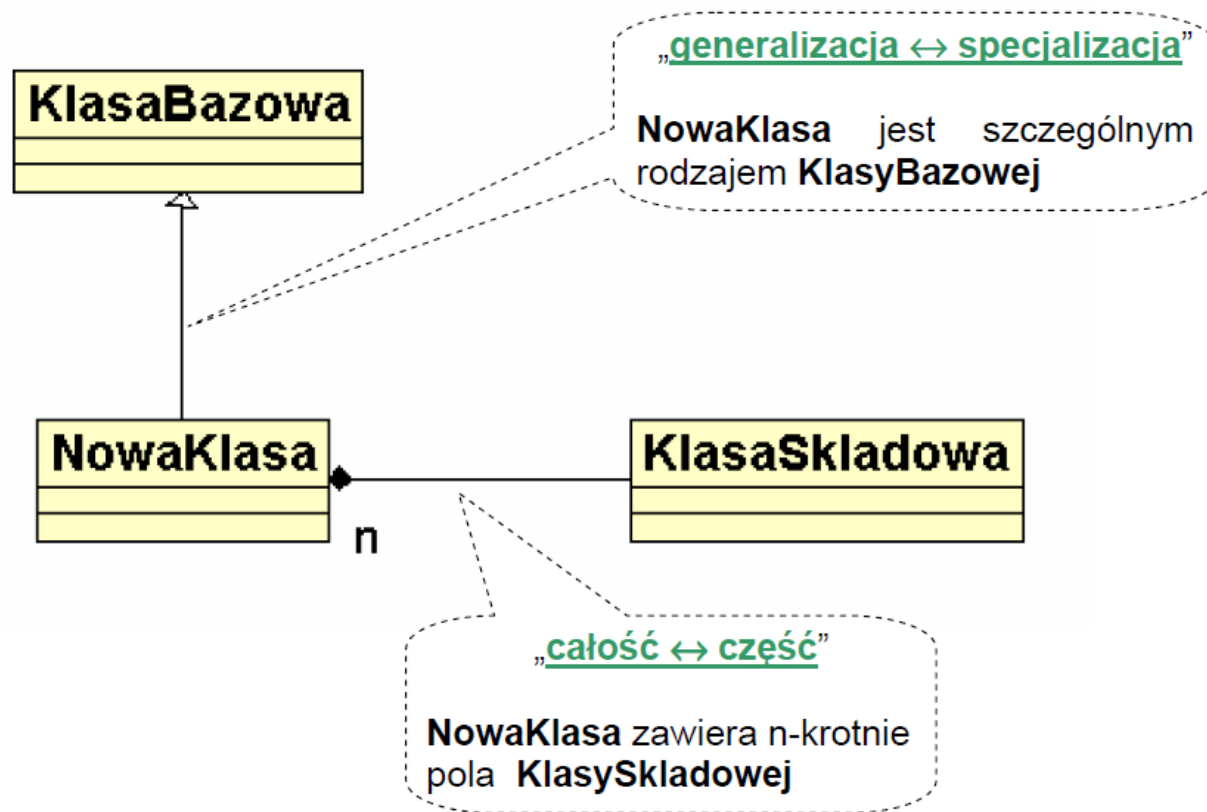
Motocykl **jest** szczególnym rodzajem Pojazdu silnikowego

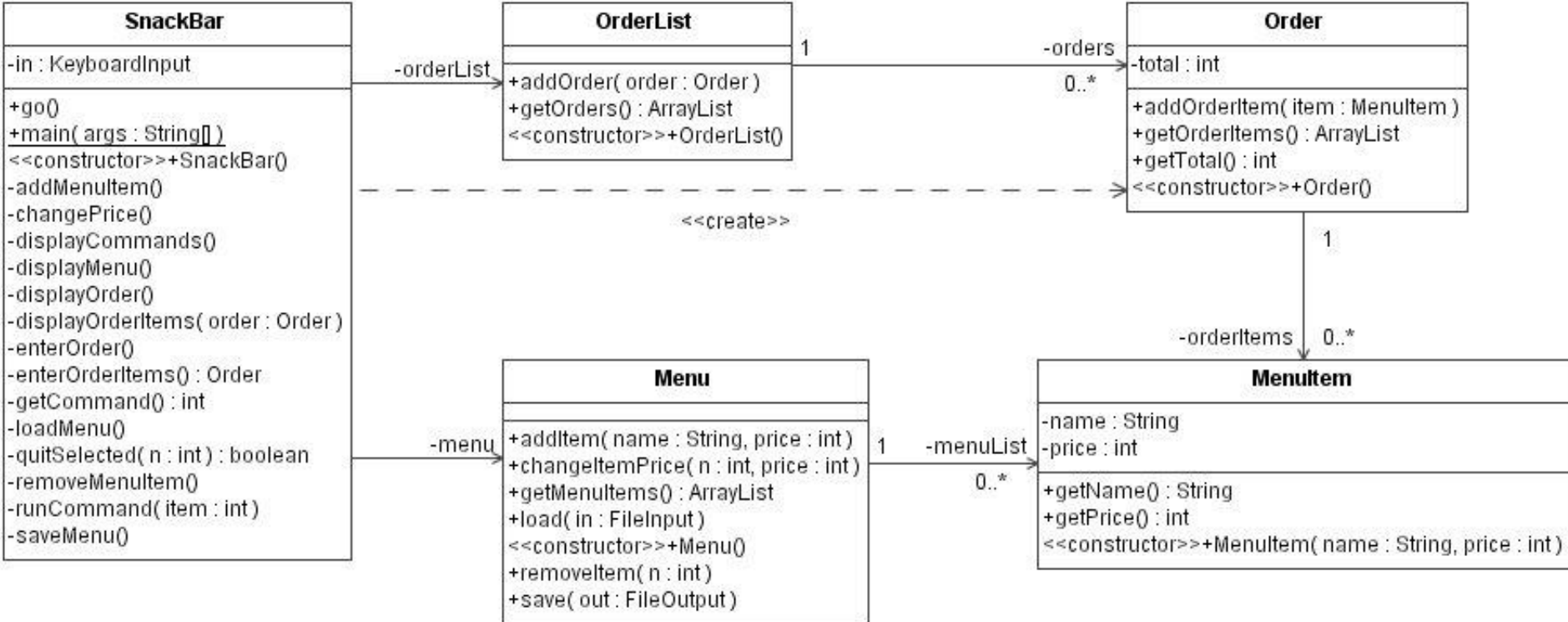
Pojazd silnikowy **zawiera** Silnik

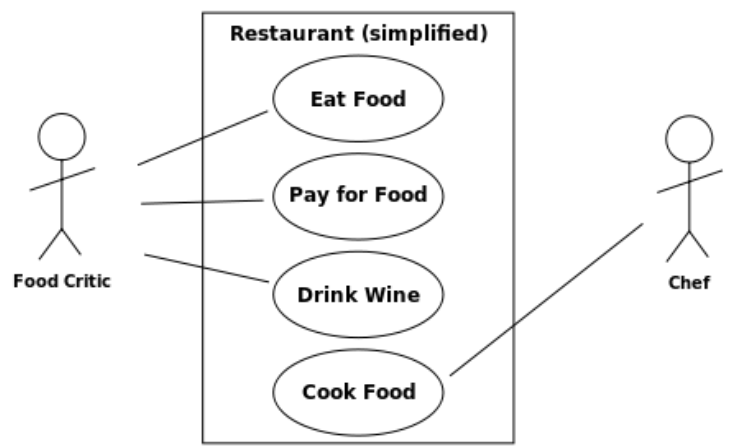
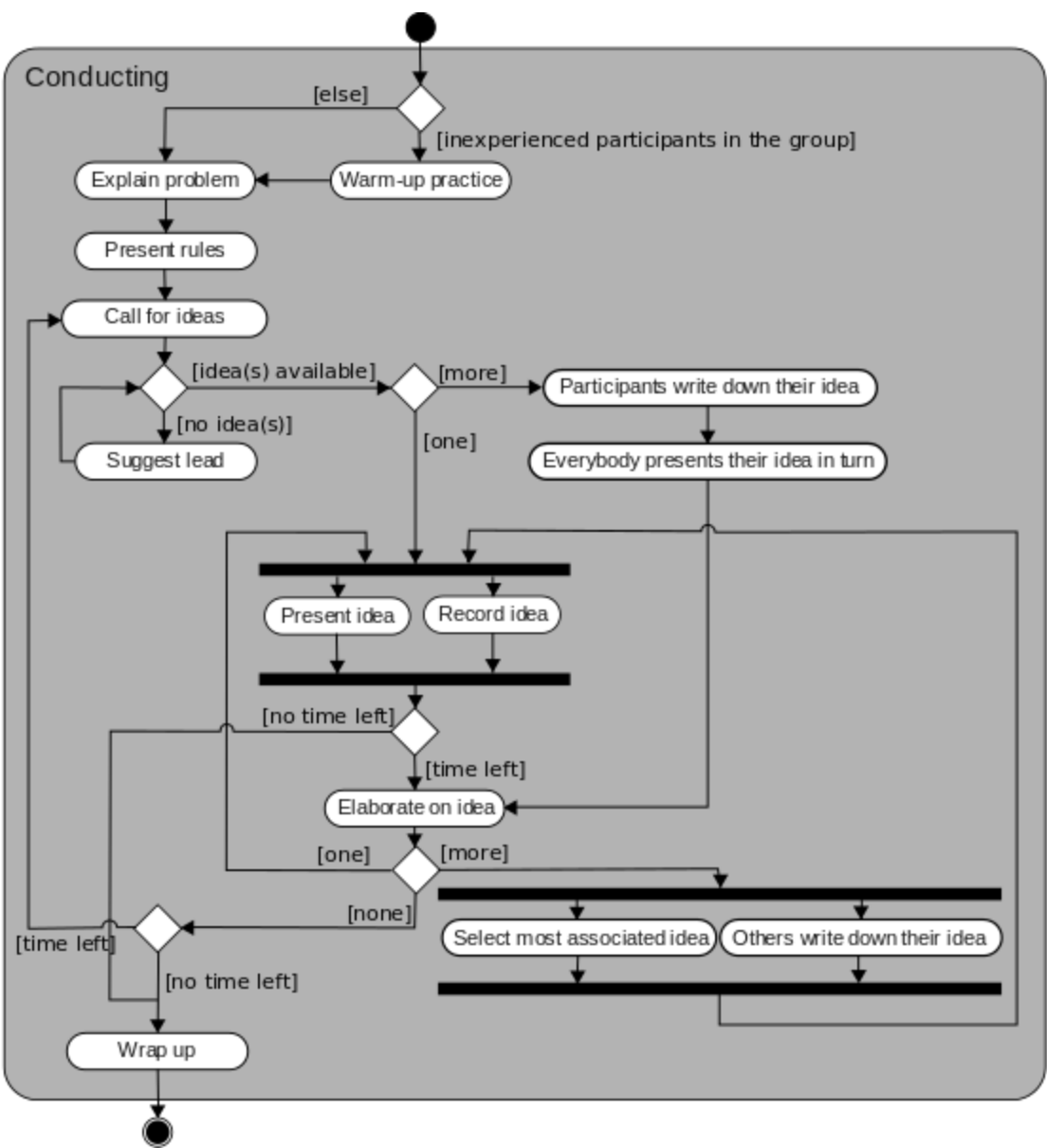
Diagramy klas w języku UML

UML (ang. Unified Modeling Language) – zunifikowany język modelowania do tworzenia systemów obiektowo zorientowanych.

Diagram klas pokazuje klasy i zachodzące między nimi relacje.







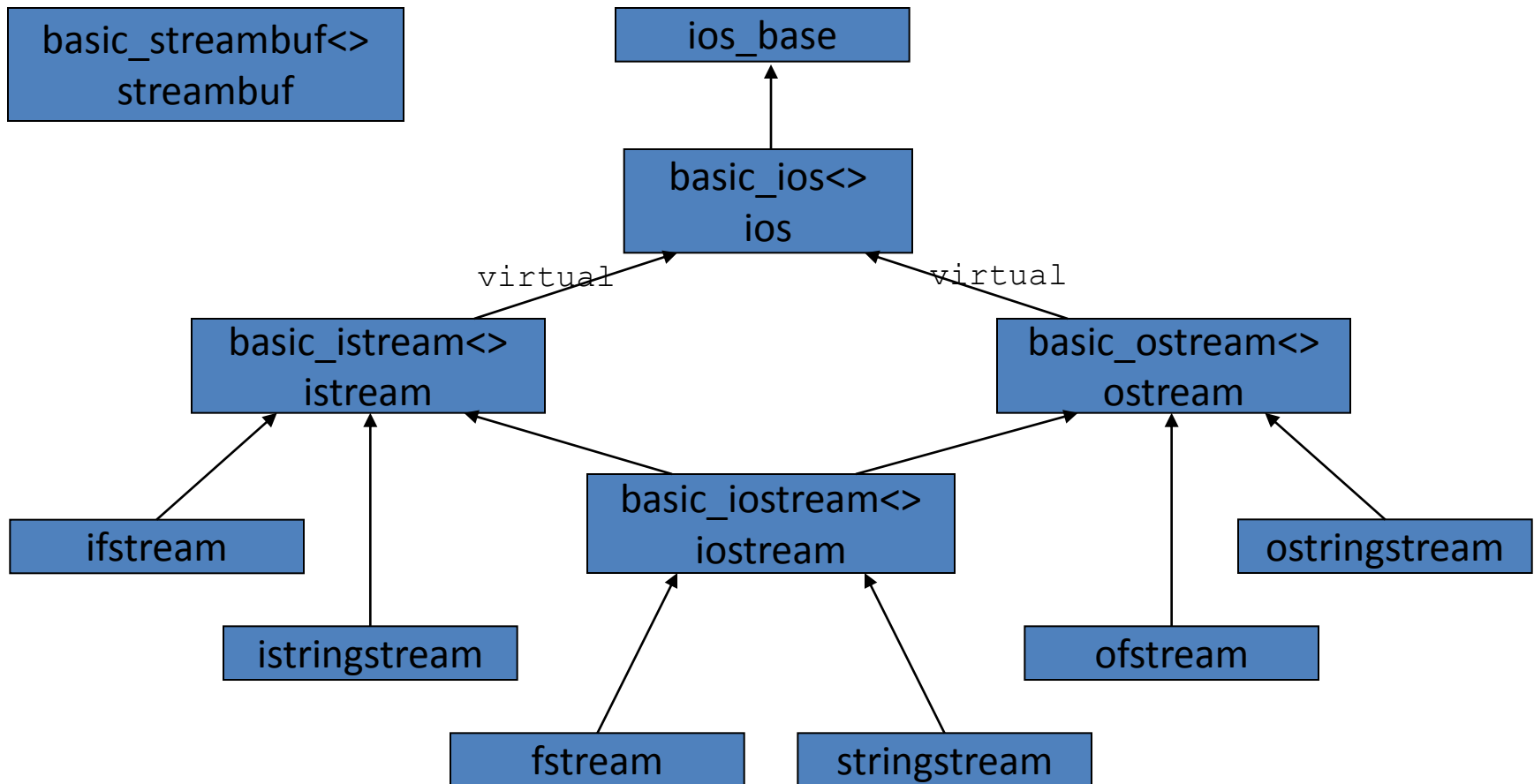
Przykład: strumienie

Strumienie

Strumień standardowy
biblioteka:
`<iostream>`

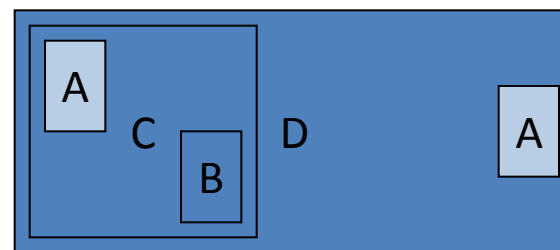
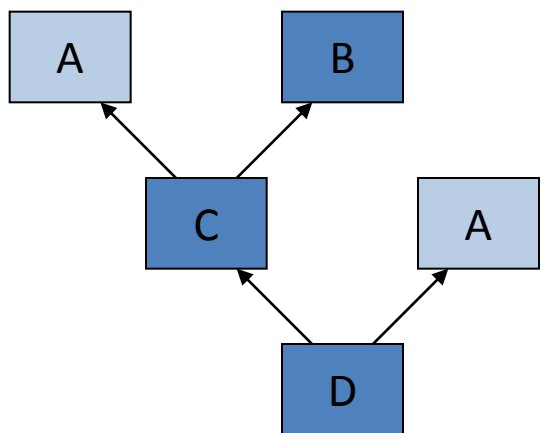


Hierarchia klas strumieniowych realizujących operacje we/wy



Dziedziczenie wielokrotne (wielodziedziczenie)

- Wielodziedziczenie może prowadzić do wielu skomplikowanych sytuacji: w pojedynczym obiekcie pewna informacja może się wielokrotnie powtórzyć.



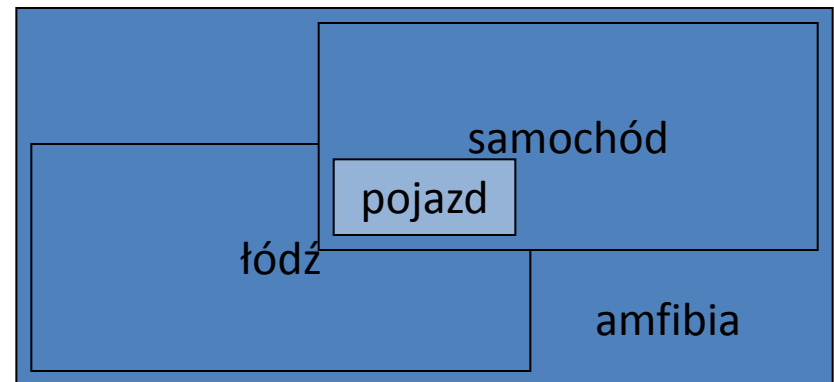
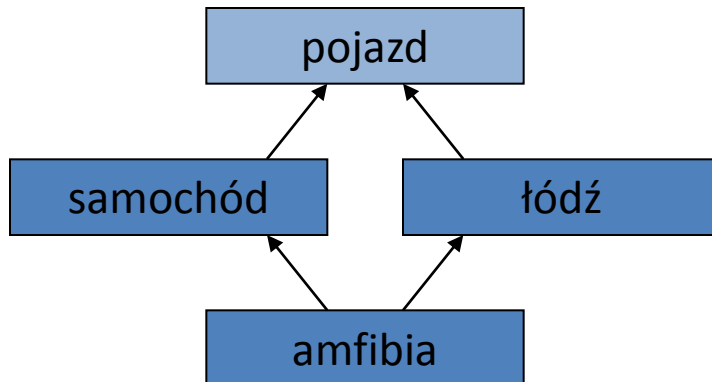
Dziedziczenie wirtualne

- Dziedziczenie wirtualne może rozwiązać część problemów z dziedziczeniem wielobazowym.
- Dziedziczenie wirtualne powoduje, że pewne informacje występujące wielokrotnie w obiekcie mogą stać się wspólne dla wielu części.
- Dziedziczenie wirtualne deklaruje się słowem `virtual` występującym na liście pochodzenia przed klasą bazową.
- Konstruktor wirtualnej klasy podstawowej jest wywoływany przed konstruktorami jej klas pochodnych.

Dziedziczenie wirtualne

- Przykład dziedziczenia wirtualnego:

```
class pojazd
    { /*...*/ };
class samochód: public virtual pojazd
    { /*...*/ };
class łódź: public virtual pojazd
    { /*...*/ };
class amfibia: public samochód, public łódź
    { /*...*/ };
```



Pobieranie danych ze strumienia

operator lub funkcja	Uwagi
>>	Operator >> domyślnie pomija białe znaki. Wczytuje ciąg znaków do pojawiania się kolejnego białego znaku
<code>get(char* gdzie, int ile, char ogran = '\n')</code>	Ogranicznik nie jest wyjmowany ze strumienia
<code>getline(char* gdzie, int ile, char ogran = '\n')</code>	Ogranicznik jest wyjmowany ze strumienia
<code>read(char* gdzie, int ile)</code>	Nie dopisuje na koncu stringu znaku NULL

Zapisywanie danych do strumienia

operator lub funkcja	Uwagi
<code><<</code>	przekierowanie do strumienia
<code>put(char znak)</code>	Wstawia do strumienia jeden znak
<code>write(char* skąd, int ile)</code>	Wstawia do strumienia określoną ilość danych

```
#include<iostream>
#include<sstream>
#include<string>

using namespace std;

// wypisanie pierwszej kolumny z 2-kol pliku
int main(int argc, char *argv[])
{
    string line;
    double x,y;
    while(getline(cin, line)){
        istringstream stream(line); // zamiana string na strumien

        stream >> x >> y;
        cout << x << endl;
    }
    return 0;
}
```

```
#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main()
{
    string a;
    cout << "Nacisnij Enter aby zakonczyc zapis.\n";
    ofstream f("log.txt");
    cin >> a;
    if (f.good())
    {
        f << a;
        f.close();
    }
    return 0;
}
```

```
#include <iostream>
#include <sstream>
using namespace std;

int main ()
{
    long x;
    string napis;
    stringstream ss;
    cout << "Podaj dowolna liczbe calkowita: ";
    cin >> x;
    ss << x;
    napis = ss.str();
    cout << "Dlugosc napisu wynosi "
         << napis.size() << " znakow." << endl;
    return 0;
}
```



```
void wypisz_date(ostream &strumien, int dzien, int miesiac, int rok) {  
    strumien << dzien << "." << miesiac << "." << rok << endl;  
}
```

```
#include <iostream>
using namespace std;

class Date
{
    int mo, da, yr;
public:
    Date(int m, int d, int y) {
        mo = m; da = d; yr = y;
    }
    friend ostream& operator<<(ostream& os, const Date& dt);
};

ostream& operator<<(ostream& os, const Date& dt)
{
    os << dt.da << '.' << dt.mo << '.' << dt.yr;
    return os;
}

int main()
{
    Date dt(5, 6, 92);
    cout << dt;
}
```