

# Języki obce

C++

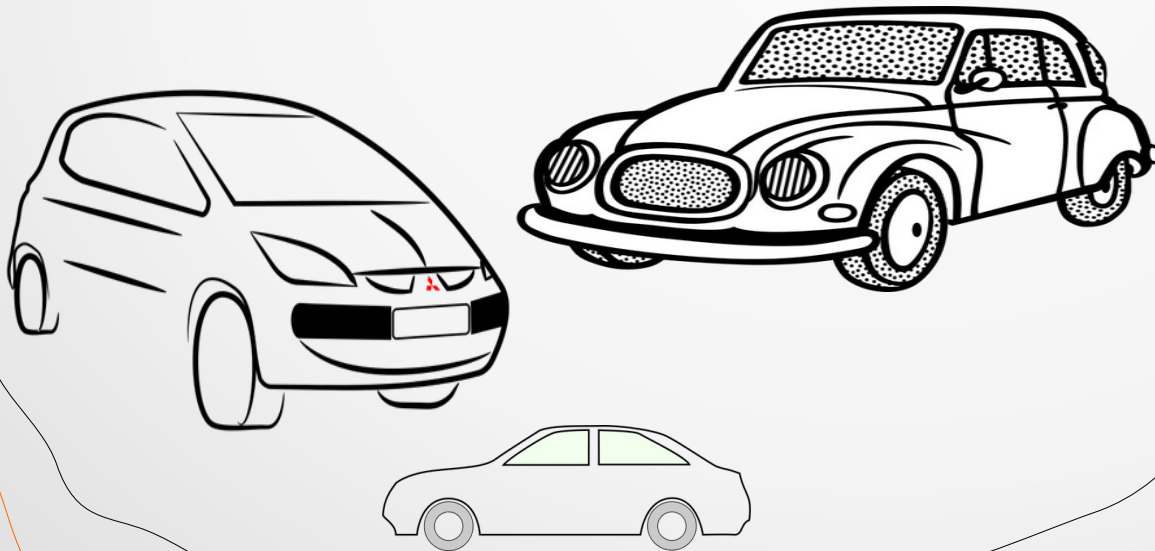
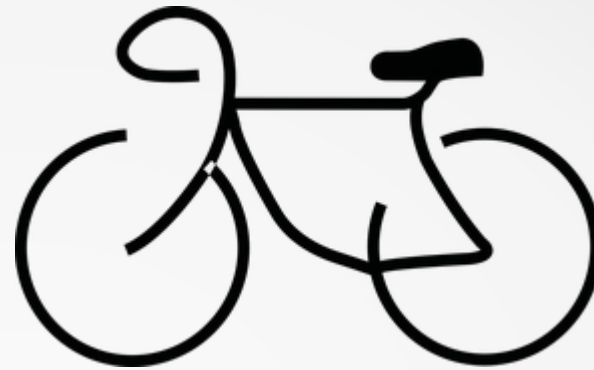
*Dziedziczenie*

*Wyjątki*

*Polimorfizm*

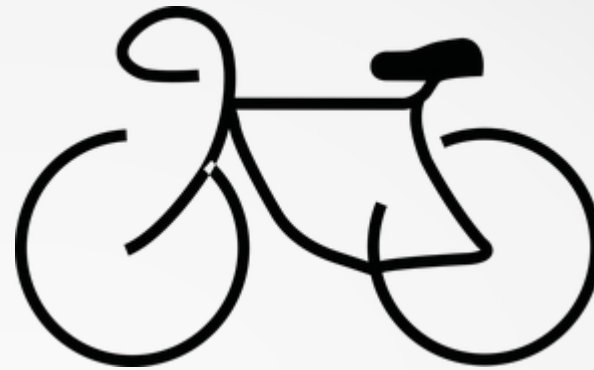
*Szablony*

# Dziedziczenie?



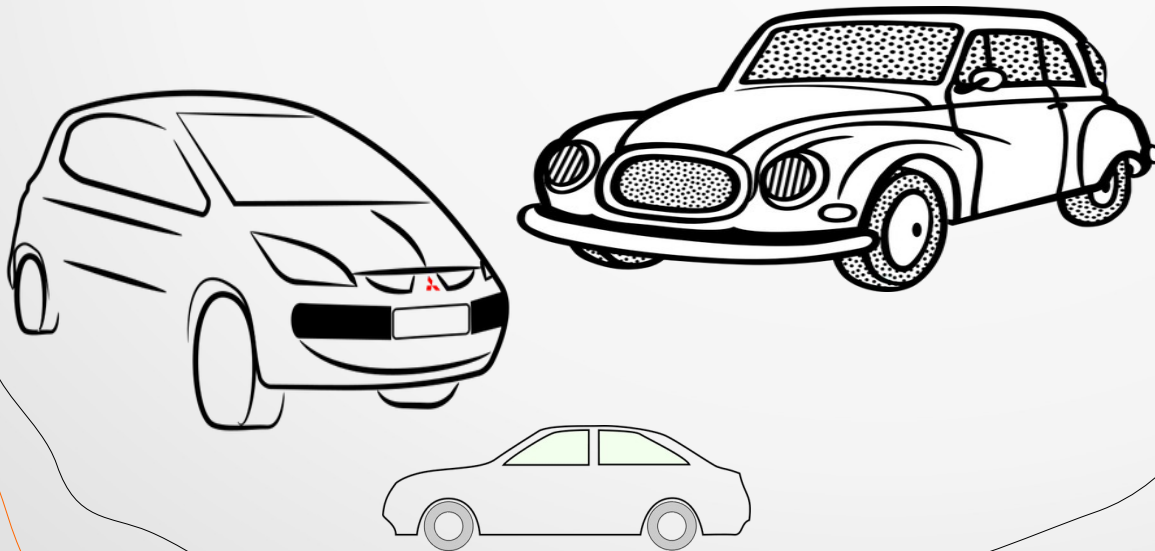
# Dziedziczenie?

Pojazdy



Rower

Samochody



# Dziedziczenie?

- Generalizacja
  - Ford, Mercedes, Fiat to samochody
- Specjalizacja
  - Ford ma coś specjalnego w porównaniu z Mercedesem i w porównaniu z Fiatem
- Po co?
  - Stopień wtajemniczenia
    - Jeśli masz prawo jazdy pojedziesz każdym samochodem
    - Jeśli wiesz gdzie jest klimatyzacja w Fordzie będziesz potrafił ją uruchomić. Jeśli nie wiesz gdzie jest w Fiacie to tego nie zrobisz
    - Nie każdy programista w zespole musi wiedzieć wszystko. Istotne jest czy wiedzą jak się komunikować i korzystać z tego co zrobili.
  - Wymusza pewien porządek

# Wyjątki

- Mechanizm który pozwala funkcjom, metodom, obiektom, że coś zadziało niepoprawnie
  - Dzielenie przez 0
  - Odwołanie się do -1 elementu tablicy
  - Przekroczenie zakresu tablicy
  - Problemy z alokacją pamięci
- Mechanizm który pozwala przechwycić tą informację i zareagować na nią w obiekcie/funkcji/metodzie, która oczekuje na wynik

# Wyjątki

Main: Oblicz wynik ilorazu?

Dzielenie: Podaj dwie liczby?

Main: 1, 0

Dzielenie: zwracam wyjątek: „dzielenie przez zero”

Main: Przechwytuje wyjątek

Main: Wyświetlam komunikat do użytkownika, ale nie zamykam programu np. okna

# Polimorfizm





# Poli? Morfizm?



Takie same elementy w kolekcji – nic szczególnego ;)



# Poli? Morfizm?



Różne elementy w kolekcji, ale każdy unikalny?

# Poli? Morfizm?



Różne elementy w kolekcji, ale każdy występujący w wielu egzemplarzach?

# Poli? Morfizm?

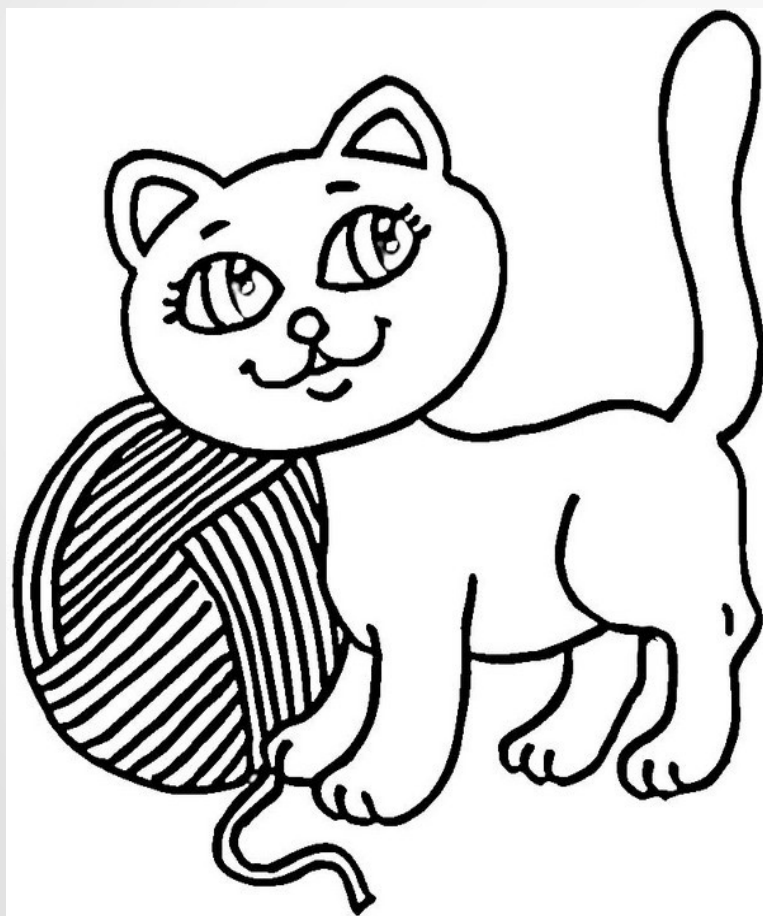
- Mamy kolekcję transformersów
  - Każdy możemy złożyć „transformować” w samochód, dinozaura, ...
  - Nauczmy nasz program wydawać rozkazy transformesom: złoż się, rozłóż się
    - Czy program musi wiedzieć jaki aktualnie trasformer jest obsługiwany?

# Poli? Morfizm?

- Mamy kolekcję transformersów
  - Każdy możemy złożyć „transformować” w samochód, dinozaura, ...
  - Nauczmy nasz program wydawać rozkazy transformesom: złoż się, rozłóż się
    - Czy program musi wiedzieć jaki aktualnie trasformer jest obsługiwany?
      - Nie, bo rozkaz jest jasny i każdy transformer rozumie co ma zrobić

Polimorfizm: umożliwia zebranie różnych typów obiektów o wspólnych cechach (np. taka sama klasa bazowa) w jedną kolekcję w taki sposób aby każdemu obiektowi wydawać rozkazy, na które każdy obiekt zareaguje typowo dla niego samego.

# Szablony





# Szablony

To samo, a jednak coś innego!



# Szablony

- Stwórzmy szablon tablicy, której elementem będzie dowolny typ np. int, double, ale też string czy jakaś nasza klasa
  - Nadal tablica, ale z innym wypełnieniem!
  - Czy znamy już jakieś szablony?



# Szablony

- Stwórzmy szablon tablicy, której elementem będzie dowolny typ np. int, double, ale też string czy jakaś nasza klasa
  - Nadal tablica, ale z innym wypełnieniem!
  - Czy znamy już jakieś szablony?
    - Tak, np. `vector<typ>`
      - vector to szablon tablicy, której elementem jest dowolny typ!