

# Obiektowy C++

**CODERS SCHOOL**

**<http://coders.school>**

- Łukasz Ziobroń, [lukasz@coders.school](mailto:lukasz@coders.school)



# About authors

## Łukasz Ziobroń

- Work at Nokia:
  - C++ software engineer @ LTE Cplane
  - C++ software engineer @ LTE OAM
  - Python developer @ LTE LOM
  - Scrum Master
  - Code Reviewer
- Trainer:
  - [Practical Aspects Of Software Engineering](#)
  - [Nokia Academy](#)
  - [Coders.school](#)
  - Internal Nokia trainings
- Occasional speaker:
  - [Academic Championships in Team Programming](#)
  - [code::dive community](#)
  - [code::dive conference](#)

# Filary obiektowości

- Abstrakcja
- Enkapsulacja (hermetyzacja)
- Dziedziczenie
- Polimorfizm



*git clone <https://github.com/LordLukin/Cars.git>*

# Abstrakcja

- Interfejs – część publiczna klasy (funkcje, pola, typy)
  - Przykład: `std::vector` na [cppreference.com](http://cppreference.com)
  - Część prywatna (implementacja) jest nieznana
- Object Oriented Design (OOD)

# Enkapsulacja

- Tryby dostępności:
  - public
  - protected
  - private
- Nienazwane przestrzenie nazw (unnamed namespace)

# Dziedziczenie

- Modyfikatory dostępu przy dziedziczeniu (tabelka)
- Kolejność wywołań konstruktorów i destruktorów
  - Najpierw klasa bazowa, potem pochodne
- **Problem diamentowy**
  - Wirtualne dziedziczenie rozwiązuje problem diamentowy

	public	protected	private
public	public	protected	private
protected	protected	protected	private
private	private	private	private

# Polimorfizm

- Funkcje wirtualne
- Funkcje czysto wirtualne (= 0)
- Klasy abstrakcyjne (min. 1 funkcja =0)
- Implementacja polimorfizmu - vtable i vptr (zaawansowane)

**CODERS.SCHOOL**

**<http://coders.school>**

