

# Programistyczny projekt funkcyjny

## Kompilator + maszyna wirtualna

Jakub Grobelny  
Łukasz Deptuch  
Kamil Michalak

## Krótki opis projektu

Naszym celem jest zaprojektowanie nowego języka programowania i napisanie jego kompilatora w Haskellu. Kompilator generował będzie *bytecode* dla naszej własnej maszyny wirtualnej, która zostanie napisana w języku C.

## Cechy języka

- Statycznie typowany język funkcyjny o ewaluacji gorliwej.
- System typów oparty na systemie Hindleya-Milnera, rozszerzony o polimorfizm *ad hoc* w postaci klas typów inspirowanych tymi z Haskellu.
  - inferencja typów
  - polimorfizm parametryczny
  - klasy typów
  - algebraiczne typy danych
  - krotki i rekordy
- Kompilacja do kodu pośredniego wykonywanego przez zewnętrzne środowisko uruchomieniowe. Pozwala to na większą przenośność skompilowanego kodu.
- Automatyczne odśmiecanie pamięci.
- Gwarancja optymalizacji rekursji ogonowej.
- Możliwość programowania imperatywnego w stylu języka *ML*.
- Możliwość definiowania własnych operatorów wraz z priorytetami i łącznością.
- System importowania modułów podobny do Haskellu.

## Zadania do wykonania

1. Wymyślenie języka.
  - Zaprojektowanie i opisanie systemu typów i semantyki języka.
  - Wymyślenie i formalne opisanie składni.
  - Zaprojektowanie systemu importowania modułów.

2. Napisanie kompilatora w Haskellu.

- Ustalenie składni abstrakcyjnej dla reprezentacji języka.
- Napisanie parsera ustalonej składni konkretnej.
- Implementacja importowania modułów.
- Wstępna implementacja systemu typów.
- Napisanie prostego interpretera i *REPL* do debugowania.
- Dodanie klas typów do systemu typów.
- Generowanie *bytecode*'u maszyny wirtualnej.
- Dodanie optymalizacji kodu wynikowego.

3. Napisanie maszyny wirtualnej w C.

- Wymyślenie odpowiedniego zestawu instrukcji.
- Zaprojektowanie formatu plików obsługiwanego przez maszynę wirtualną.
- Napisanie interpretera instrukcji.
- Implementacja odśmiecania pamięci.
- Zaprojektowanie i zaimplementowanie *Foreign Function Interface* w celu umożliwienia wywoływania funkcji napisanych w C.

4. Dalsze rozwijanie języka oraz kompilatora.

- Rozszerzenie systemu modułów o importowanie skompilowanych plików zamiast jedynie kodu źródłowego.
- Automatyczne wyprowadzanie instancji pewnych klas typów przez kompilator.
- Napisanie prawdziwego *REPL*.
- Dodawanie różnych rozszerzeń systemu typów zaczerpniętych z Haskella.
- Polepszanie jakości błędów kompilacji.

5. Dalsze rozwijanie maszyny wirtualnej.

- Współbieżność.