

Kurs języka Ruby

Lista 4.

Zadanie 1. Typowym zadaniem na zajęciach z programowania jest implementacja wypożyczalni (płyt, samochodów etc) czy biblioteki. Zazwyczaj funkcjonalności czy schematy takich programów są bardzo podobne. Zdefiniuj więc moduł wypożyczeń **Wypozyczenia** (lub kilka odpowiednich modułów) implementujący odpowiednie schematy wypożyczenia i zwracania przedmiotów do wypożyczalni. Moduł ten powinien być tak napisany, aby łatwo było go *wmiksować* do odpowiednich klas by utworzyć prawdziwą wypożyczalnię.

Korzystając z tego modułu zaimplementuj klasę **Biblioteka** korzystającą z **Wypozyczenia** implementującą wypożyczalnię książek.

Zadbaj o personalizację komunikatów, tj. aby komunikaty były postaci „wypożyczono książkę ...” a nie „wypożyczono obiekt #<Ksiazka:0x7fd04e280f20>”, o ile oczywiście przykładową klasą będzie biblioteka.

Zadanie 2. Ważnym choć czasem niedocenianym elementem rozwijania oprogramowania jest testowanie. Obiekty można na przykład testować dodając odpowiednie metody testujące. Dla wygody można przyjąć, że nazwy metod testujących zaczynają się od **test_**.

Rozszerz podany na wykładzie mix-in **Debug** o procedurę **check**, która wyszukuje w klasie wszystkie metody zaczynające się na **test_** i wykonująca je. Przyjmij, że metody **test_*** zwracają wynik testu jako obiekt klasy **String**.

Zadanie 3. Zaprogramuj klasę **DrzewoBinarne** implementującą drzewo poszukiwań binarnych wraz z operacjami **wstaw**, **istnieje?** i **usun**. Przyjmij, że elementy drzewa są obiektami klasy **Element**. Podaj przykład wykorzystania tych klas. Zwykle do implementacji takiej klasy konieczne są operatory **>**, **<** czy **==**, jednak zamiast tego wygodniej jest zaimplementować tylko metodę **<=>** i dołączyć *mix-in* **Comparable**.

Zaprogramuj **StringBT** jako podklasę **DrzewoBinarne** przechowującą obiekty klasy **String**.

Każde zadanie jest warte 3 punkty. Na pracowni oddaje się jedno zadanie.

Marcin Młotkowski