

Programowanie obiektowe

Lista 8.

Poniższe zadanie jest do zrobienia w języku Ruby. Można za nie otrzymać 8 punktów.

Na tej liście będziemy implementować dwuosobową grę w karty.

Do gry będzie potrzebna klasa obiektów reprezentujących standardowe karty¹, gdzie obiekt reprezentuje pojedynczą kartę. Klasa ta powinna implementować porównanie kart zgodne z zasadami gry. Klasa ta powinna również implementować metodę `to_s`, która zwraca reprezentację karty w postaci tekstowej, np. `Karta.new(2, :kier).to_s` powinna zwrócić np. "2♥". W praktyce implementacja 52 kart jako odrębnych klas może być uciążliwe i niepraktyczne, dlatego można zaprogramować jedną klasę **Karta**, a kolory i figury będą odpowiednimi atrybutami. Zaprogramuj też odpowiedni mechanizm chroniący przed utworzeniem dwóch obiektów reprezentujących tę samą kartę. Można to zrobić np. poprzez implementację metody statycznej klasy **Karta**, która będzie "wydawać" kolejne karty (trochę podobnie jak we wzorcu *Singleton*). Zaprogramuj też odpowiednią metodę tasowania talii.

Gra będzie się toczyć pomiędzy dwoma graczami (może być więcej, ale wystarczy dwóch). Zaprogramuj dwie klasy graczy:

- pierwsza klasa będzie służyć jako prosty interfejs konsolowy do komunikacji z użytkownikiem. To użytkownik wpisując odpowiednie polecenia będzie brał udział w rozgrywce;
- druga klasa będzie implementować prostą strategię gry, może być nawet losowa.

Klasy nie powinny być powiązane dziedziczeniem (za wyjątkiem standardowego dziedziczenia po klasie **Object**), natomiast powinny implementować te same metody, aby można je było stosować zamiennie. Każda z tych klas powinna posiadać własną kolekcję kart, które będzie wykładać podczas rozgrywki. Wygodne jest też nadawanie nicków poszczególnym graczom.

Zaprogramuj również klasę **Gra**, która będzie implementować zasady gry w karty. Jej rolą jest

- pamiętanie listy graczy;
- rozdanie kart z potasowanej talii;
- wskazywanie gracza, który ma wykonać ruch;
- pamiętanie listy wyłożonych kart;
- wskazywanie zwycięzcy lub remisu.

Taka implementacja powinna umożliwiać rozgrywki typu *człowiek-komputer* czy *komputer-komputer* inicjując obiekt **Gra** obiektami graczy z różnych klas.

Można wybrać dowolną grę, np. *wojna* czy *oczko*. Można też zaprogramować grę wymagającą innej talii niż standardowa.

Te zadania będą kontynuowane na następnej liście zadań.

Marcin Młotkowski

¹powszechnie używa się tzw. kart talii typu francuskiego