Dr Jakub Jernajczyk

Akademia Sztuk Pięknych
im E. Gepperta we Wrocławiu

Cyfrowy idealizm?

O podobieństwach pomiędzy projektowaniem wizualnym, paradygmatem programowania obiektowego oraz Platońskim idealizmem.

Wybrana literatura problemu:

1994: B. Stroustrup, *Język C++*

2000: D. Rayside, G. T. Campbell, *An Aristotelian Understanding of Object-Oriented Programming*

2002: R. Janusz, *Program dla Wszechświata. Filozoficzne aspekty języków obiektowych*

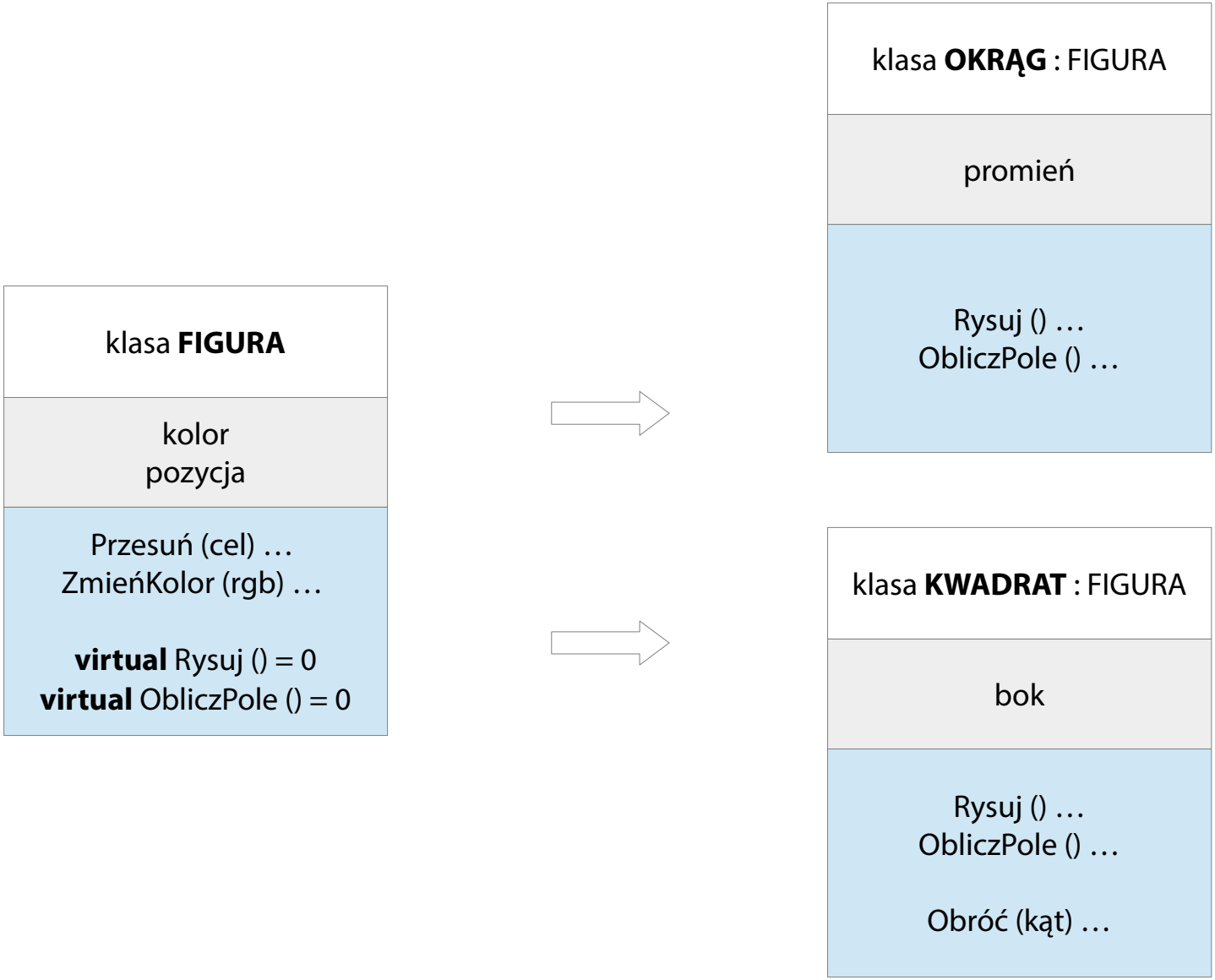
2006: R. Giguette, *Building objects out of Plato: applying philosophy, symbolism, and analogy to software design*

2008: J. Jernajczyk, *Elementy filozofii w sztuce mediów dyskretnych* [maszynopis]

2016: W. Tylman, *Computer Science and Philosophy: Did Plato Foresee Object-Oriented Programming?*

Główne cechy paradygmatu programowania obiektowego:

- hermetyzacja (enkapsulacja, kapsułkowanie)
- definiowanie abstrakcyjnych typów obiektów – klasy (zazwyczaj)
- dziedziczenie – nadklasy i podklasy
- hierarchia klas – klasy podstawowe i pochodne
- metody wirtualne, polimorfizm


klasa **FIGURA**
kolor
pozycja
Przesuń (cel) …
ZmieńKolor (rgb) …
virtual Rysuj () = 0
virtual ObliczPole () = 0
klasa **OKRĄG** : FIGURA
promień
Rysuj () …
ObliczPole () …
klasa **KWADRAT** : FIGURA
bok
Rysuj () …
ObliczPole () …
Obróć (kąt) …

B. Stroustrup, *Język C++*

Jeśli myślisz o „tym” jako o osobnej idei, to uczyń z tego klasę.
Jeśli myślisz o „tym” jako o osobnym elemencie, to uczyń z tego obiekt pewnej klasy.

…

Język C++ zaprojektowano jako dobre narzędzie budowy wielu różnych systemów i do bezpośredniego wyrażania wielu różnych idei.

Platon, *Timaios*, 52

… istnieje jeden rodzaj rzeczy, niezmienny, niezrodzony i nieginący,
który ani w siebie nie przyjmuje niczego skądinąd, ani sam w nic innego
nigdzie nie przechodzi, niewidzialny i w żaden sposób niedostrzegalny
– oglądać go może tylko myśl rozumna. I drugi rodzaj rzeczy, nazywany
tak samo i podobny do tamtego, spostrzegalny, zrodzony, zmienny ustawicznie,
który powstaje w pewnym miejscu i znowu stamtąd przepada – uchwycić
go potrafi mniemanie i spostrzeżenie.

Paradygmat programowania obiektowego a Platońska *teoria idei* (*form*):

- *klasy* jako ***idee*** oraz *obiekty* jako ***rzeczy***
- wielość i zmienność rzeczy/obiektów a niezmienność i jedność idei/klas

Pojęcie «klasy» w programowaniu obiektowym jest Platońskie w tym sensie, że w kontekście działania programu klasy istnieją przed obiektami (tak jak idee preegzystują względem rzeczy materialnych) oraz są używane jako wzory do wytwarzania obiektów.

(Rayside & Campbell, 2000)

Hierarchiczna struktura rzeczywistości u Platona:

- świat pierwszych i najwyższych zasad (*Jedno* i *Diada*)
- świat idei:
    - liczby i figury idealne
    - idee najogólniejsze – metaidee
    - idee szczegółowe
- świat bytów matematycznych
- świat zmysłowy

Analogia do hierarchii klas – poziom niższy nie może istnieć bez wyższego; można usunąć to co zależne, ale nie odwrotnie.

Metafizyka Arystotelesa:

- parze *klasa – obiekt* lepiej odpowiadać może para *forma – synolon* (konkretne połączenie formy i materii)

Możemy powiedzieć, że klasy w programowaniu obiektowym są Arystotelesowskie w takim sensie, że programista w obszarze danego zagadnienia ogarnia najpierw elementy jednostkowe (obiekty) a dopiero potem opracowuje zawierające je pojęcia abstrakcyjne (klasy).

(Rayside & Campbell, 2000)

Dwa modele programowania obiektowego:

- oparte na klasach (C++, C#, PHP, Java) – bliższe Platonowi
- oparte na prototypach (JavaScript, Python) – bliższe Arystotelesowi

Mechanizm dziedziczenia a system Plotyna:

- wyłanianie się (*emanacja*) kolejnych poziomów bytu (*hipostaz*) z zasady najwyższej (*Jednego*)

- Jedno to coś *absolutnie prostego, co jest racją bytu tego co złożone i mnogie.* Prostota Jednego *nie jest uboga* – Jedno jest *nieskończoną mocą*.

- *Wszystkie byty są bytami dzięki jedności* (*Enneady*, VI, 9, 1)

Jedno $\xrightarrow{j.}$ Umysł $\xrightarrow{j.}$ Dusza $\xrightarrow{j.}$ byty fizyczne

Uwaga: Umysł i Dusza nie są tu tożsame z ludzkim umysłem oraz duszą; dla Plotyna oznaczają one odrębne poziomy bytu.

Podział języków obiektowych (jeden z wielu możliwych):

- brak wspólnego korzenia dla wszystkich klas/obiektów
  np. C++, PHP

- istnieje jeden korzeń – nadklasa dla wszystkich klas/obiektów
  np. C#, Java, Ruby

Podobieństwa do idealizmu i obiektowości na poziomie narzędziowym:

- przed-filozoficzne, praktyczne intuicje: wzór i obraz, oryginał i kopie, matryca graficzna i odbitki, formy przemysłowe, schematy

- wizualne narzędzia cyfrowe do projektowania i edycji grafiki:
  – tworzenie ogólnych wzorców oraz zależnych od nich instancji
  – możliwość modyfikacji struktury wewnętrznej wzorców
  – przykładowe narzędzia: Adobe After Effects, Toom Boom Harmony, Adobe Premiere, SketchUp, Adobe Director

- prezentacja środowiska Adobe Flash (obecnie Animate)

Omówienie rozważanych analogii w środowisku Adobe Flash (Animate):

- enkapsulacja – zamykanie w symbolach własności i zachowań
- dualizm *symboli* i ich *instancji* (w analogii do par: klasa–obiekt, idea–rzecz)
- wiele instancji tego samego symbolu (różnice zewnętrzne)
- zależność instancji od symboli (istnienie oraz zmiana cech wewnętrznych)
- mechanizmy dziedziczenia (powielanie symboli)
- dwa podejścia: „Platońskie” (projektowanie symboli) i „Arystotelesowskie” (przekształcanie na symbole)

Dyskusja dotycząca możliwych źródeł rozważanych analogii:

- bezpośrednia i świadoma inspiracja
- inspiracja nieuświadomiona (Platonizm obecny w kulturze)
- całkowicie przypadkowa zbieżność podobnych koncepcji
- oddziaływanie podobnych przed-naukowych koncepcji na różne dziedziny ludzkiej aktywności, w różnych okresach