A Solution to the Square-Rectangle Problem Within the Framework of Object Morphology

Zbyněk Šlajchrt1

1 Department of Information Technologies, Faculty of Informatics and Statistics,

University of Economics, Prague

W. Churchill Sq. 4, 130 67 Prague 3

zslajchrt@gmail.com

**Abstract:** Insert abstract in English here. Please follow the instructions for authors and respect maximum abstract length, which it should be between 100 – 200 words in length. Do not cite references to literature in the abstract.

**Keywords:** Max seven keywords or phrases state here, separated by commas.

# Introduction

The square-rectangle problem (SRP) exposes a couple of flaws inherent to object-oriented programming (OOP). In particular, the problem is closely related to subtyping and inheritance and manifests itself as a violation of the Liskov substitution principle (LSU).

To solve SRP, one attempts to determine the proper subtyping, resp. inheritance relationship between the rectangle and the square. In mathematics, a square is a degenerated version of a rectangle having the same width and height. It follows that a square **is a** rectangle and thus the “IS-A” relationship should be used to model the relationship between squares and rectangles; in other words the square inherits from the rectangle (or the square is a subtype of the rectangle).

If the rectangle class exposes methods for setting the width and height of a rectangle then also the square class automatically inherits those methods. Given that according to the Liskov substitution principle squares are substitutable for rectangles, any square may be used in any routine processing rectangles, including the routines modifying the state of the processed rectangles using the public width and height setter methods. However, in case those setters are invoked on a square the resulting state of the square may be in violation of the constraint stipulating that the width and the height of a square must be the same.

This problem exemplifies a more general modeling problem, in which a class hierarchy consists of a base class and of subclasses that are constrained versions of the base class. The point here is that some methods of the base class can modify the object’s state so that the modified state violates the constraints of a subclass. Such a modeling situation may occur quite often and may lead to serious design-related issues in the final application as long as the problem is not identified and solved early in the modeling phase of the development.

There are a number of possible solutions of this problem and some of them are described in the literature review section. Broadly speaking, all those solutions do not solve the problem comprehensively and often introduce secondary problems.

The purpose of this paper is to present a novel approach to the above-mentioned problem based on a conceptual framework called object morphology (OM), which is being developed by the author’s doctoral thesis [X]. In OM an object is in principle a mutable entity, while the mutation does not affect the object’s state only, but also its type. There are no class or type hierarchies in OM; instead, objects are modeled by the so-called morph models describing permitted structural mutations of the objects.

The presented approach entirely shifts the view on the constraint violations. In this view, a violation of constraints caused by the mutated object’s state is no longer considered a faulty situation; instead the violation is simply interpreted as a signal to mutate the object to a different form (type), with which the mutated state does not conflict. To put it in the context of SRP, a violation of the square constraints caused by setting the width and the height properties to different values would lead to mutating the square instance into a rectangle, while preserving the identity of the object.

The solution is demonstrated on an extended version of SRP using Morpheus, a proof-of-concept implementation of OM in the Scala language.

This paper focuses on the conceptual side of the problem, leaving aside other important aspects of the solutions, such as performance, mainly on account of the fact that Morpheus is still in the proof-of-concept stage.

# Literature review

weakening the contract for the modifiers, workarounds, avoiding mutability of classes, factoring out modifiers to other classes, inverting inheritance and others.

# Research method

Constraint-enforcing specializations are not in general equivalent to subtypes in the sense of LSU and .

Specialization, which puts constraints on the specialized classes, is often mistaken for subtyping/inheritance/derivation, which adds new attributes and new or overriding methods only, such as the Animal-Cat relationship, for instance.

Animal/Cat vs. Rectangle/Square

Using OCL for constraints, OCL-based morphing strategies

# General information and styles

**This template is intended only for the contribution written in English! For Czech or Slovak written contribution, use the appropriate template.** Please use the following predefined formatting styles (Heading 1, Heading 2, main text, etc.) in this template.

Correctly formatted article according to this template should look like this:

[**EXAMPLE IN PDF**](http://aip.vse.cz/download/aip-template-en-example.pdf)

Articles can be published in English, Czech and Slovak. Articles are accepted only through [the journal publishing system](http://aip.vse.cz/index.php/aip/login) (if you have any questions, please contact the editor). Author will be informed about the acceptance of the paper by e-mail. The following communication (regarding the review process) goes on by the e-mail.

The minimum extent for the acceptance of the contribution as an original scientific paper (**Peer-reviewed papers section**) to the peer-review procedure is 4 pages and maximum is 20 pages. If you are interested in publishing longer paper, you have to inform the editor first. For contributions specified to the **section Peer-reviewed papers** is recommended this structure:

* Introduction
* Literature review and research methods
* Solutions and Results
* Discussion
* Conclusion
* Acknowledgement
* References

Parts can be structured into multiple subsections.

**Contributions to the non-reviewed sections** (Book reviews, Miscelanea and Reflections) at the end of the journal must be in the range from 2 to 6 pages.

# Heading 1 for the title of sections

Normal for the text of the paper.

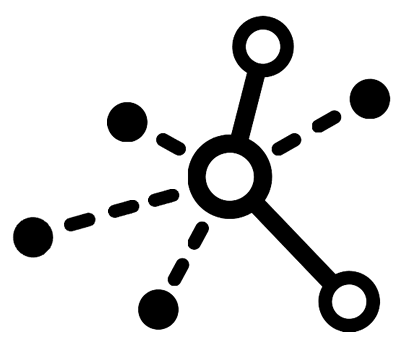
* Bullet A
* Bullet B
  + Bullet 1
  + Bullet 2
* Bullet C

## Heading 2 for the sub-titles of sections

### Heading 3

## Figure

All figures and tables must be centred and formatted according to the following examples.



**Fig. 1.** Caption for the captions of figures. Source (Berg, 2014, p. 57).

## Table

|  |  |  |
| --- | --- | --- |
| **AAA** | **BBB** | **CCC** |
| Aaa1 | Bbb1 | Ccc1 |
| Aaa2 | Bbb2 | Ccc2 |
| Aaa3 | Bbb3 | Ccc3 |
| Aaa4 | Bbb4 | Ccc4 |

**Tab. 1.** Caption for the captions of tables. Source (Berg, 2014).

## Source Code

#include <iostream>

using namespace std;

#define PI 3.14

int main ()

{

double r=4.0; // radius

double circle;

circle = 2 \* PI \* r;

cout << circle;

return 0;

}

## Equation

|  |  |
| --- | --- |
|  | **(1)** |

To create equations use the integrated editor "Equation editor" in MS Word. Equation insert in invisible table and each equation should be numbered.

# References

The contribution should primarily refer to the relevant scientific journals and conferences that are indexed in the Web of Knowledge and Scopus. References have to be **alphabetically sorted.** For links to references in the text, use the following examples using the author's surname and year of publication:

* **1 author** – Berg (2014, pp. 55-57) pointed out the problem… The problem is well-known (Magel, 2013a, p. 47) and further expanded in other papers (Lateg, 2013; Margel, 2013b; Apple, 2012).
* **2-3 authors** – Joergen a Jones (2009) improved used methods… Nevertheless in practice is used CUW method (Kang, Tucin & Kent, 2002).
* **More than 3 authors** – Skálová et al. (2010) provide the solution… The solution already exists (King et al., 2014).

**Examples of list of references**

To cite sources in the part of **References** use APA style (6th edition). The tutorial can be found here: <http://www.apastyle.org/learn/tutorials/basics-tutorial.aspx> In case of problems or doubts, we recommend to use a generator for APA citations on the internet. Below are examples of the most commonly used sources:

***Book:***

Berg, S. (2014). *Services marketing management*. London: Routledge.

Magel, J. (2013a) *Mastering data mining*. (1st ed.) Stockholm: Stockholm School of Economics.

Magel, J. (2013b) *Mastering GUHA. Stockholm*: Stockholm School of Economics.

***Article in journal:***

Joergen, P., & Jones, K. (2009). Random data analysis and measurement procedures. *Journal of Systems Integration*, 5(2), 55-85. Retrieved from <http://si-journal.org/joergen-jones-2009.pdf>

Kang, O., Tucin, J., & Kent, K. (2002). CUW methodics for marketing management. *Journal of Information Management*, 54(3), 1502-1535. doi: 10.7160/jim.2002.06784

***Article in other periodical (newspaper):***

Lateg, R. (2013, March 25). A survey of data provenance in e-science. *New York Review*, pp. 8-11.

***Conference contribution or chapter of the book:***

Skálová, U., Hopstal, H., Kuruc, T., & Krebs, W. (2010). The role of anomalous data in knowledge acquisition. In J. Jicinsky & P. Trejbal (Eds.), *5th International Conference on Informatics* (pp. 248-310). Kaunas: Walter Verlag.

King, U., Smith, L., Jones, E., Kain, W., & Wright, V. (2013). Power-law distributions in empirical data. In J. Rais (Ed.), *The design and analysis of spatial data structures* (1 ed., pp. 45-122). London: Research Life.

***Reference to the web page or e-document:***

Anders, Q. (2014, April 12). *Most-trusted brands*. Retrieved from <http://googleblog.blogspot.cz/anders-brands-2014>

Apple. (2012). *CFNetwork Programming Guide*. Retrieved from <https://develo-per.apple.com/library/mac/documentation/Networking/Conceptual/CFNetwork/CFNetwork.pdf>.