



TECHNICAL UNIVERSITY OF CLUJ-NAPOCA

ACTA TECHNICA NAPOCENSIS

Series: Applied Mathematics, Mechanics, and Engineering
Vol. 63, Issue I, March, 2020

IMPROVING DESIGN OF A TRIANGLE GEOMETRY COMPUTER APPLICATION USING A CREATIONAL PATTERN

Anca-Elena IORDAN, Florin COVACIU

Abstract: *In the process of developing a computer application, design represents probably the most important factor that can affect its quality. Propitiously, there are currently several design patterns whose use can provide a solution of a general problem. Design solutions of competent architects are represented through the design models. Thereby, any design model aims to solve a specific design issue in case of object oriented programming paradigm. This paper presents an interactive computer application that can be used in the study of triangle geometry. The design of the computer application is improved by using of the "Factory" creational pattern, and the implementation is accomplished through the Java programming language.*

Key words: *Triangle Geometry, Design Patterns, Factory, UML, Java.*

1. INTRODUCTION

Due to the increasing demand for software lately, we are pursuing a development of software paradigms so that they improve the quality and productivity of computer application development. The design pattern [1] is a universal answer to an issue that frequently occurs in computer application design, but not a final design that can be transformed directly into code. The use of patterns guarantees the creation of transparent structures that allow easy understanding, extension and reuse of software. The description of the design patterns [2] provides information about the structure, the roles of the participant, the interaction between the participants and, in particular, the intention for which they should be used.

In this paper, we present the stages of developing to an interactive software in the field of triangle geometry. During the design stage, the variant without the use of patterns will be presented, as well as the variant that uses the "Factory" creational pattern [3]. This type of model instantiates objects which does not present outside the creative reasoning and will report to the new object instantiated using the interface. In order to show the improvement of the design by using this patterns it was chosen as

the subject the triangle geometry because this domain has multiple applications in the engineering field, such as electrical [4-5], electronics [6], mechanical [7] and robotics [8-9].

2. TRIANGLE GEOMETRY COMPUTER APPLICATION

The triangle geometry was implemented using Java programming language [10]. The graphical user interface was structured in two parts: a theoretical presentation part and a simulation part. Moving figures on the screen help users to grasp the meaning of mathematical ideas intuitively. From the main application window it can be selected by a main menu the following options:

- Important points in a triangle [11]: incenter, circumcenter, orthocenter, centroid, Lemoine point, Nagel point, Gergonne point, Vecten point.
- Important lines in a triangle [12]: medians, altitudes, midlines, symmedians, Euler line, Nagel line.
- Circles associated with a triangle: incircle, circumcircle, nine-point circle, Lemoine circle.

The computer application design shown in figure 1 is difficult to decipher. The diagram which showing the structure of the classes instantiates many duplicates as manifold object of “Point2D” class. Hence, creational design model will represent the optimal answer in the efficient management of the code.

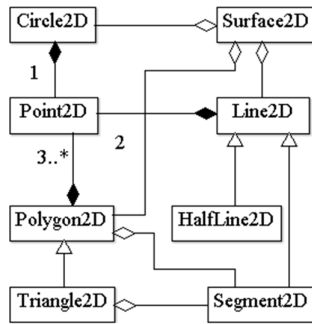


Fig. 1. UML class diagram

3. IMPROVING THE PROPOSED DESIGN USING FACTORY PATTERN

Factory template [13] is part of the most used models in computer systems design, being classified as a creational template, therefore a template that involves the creation of class instances. Factory template instantiates variable which does not present outside the logic reasoning.

“GeometricElement2D” class implements factory template and, when various classes, like “Point2D”, “Segment2D”, “Triangle2D”, “Line2D” or “Circle2D”, must be instantiated, it will preclude the source code in these derived classes from being rewritten all over the various classes. The UML diagram corresponding to the classes with factory template is presented in figure 2.

The “GeometricElement2D” was introduced as an abstract class. A “draw” method was introduced in this abstract class. Afterwards, some derived classes like “Point2D”, “Line2D”, “Polygon2D” and “Circle2D” will acquiring to it. Forwards, “Geometric-Element2DFactory” class was designed on the principle of the factory template and the function “getGeometricElement” was created.

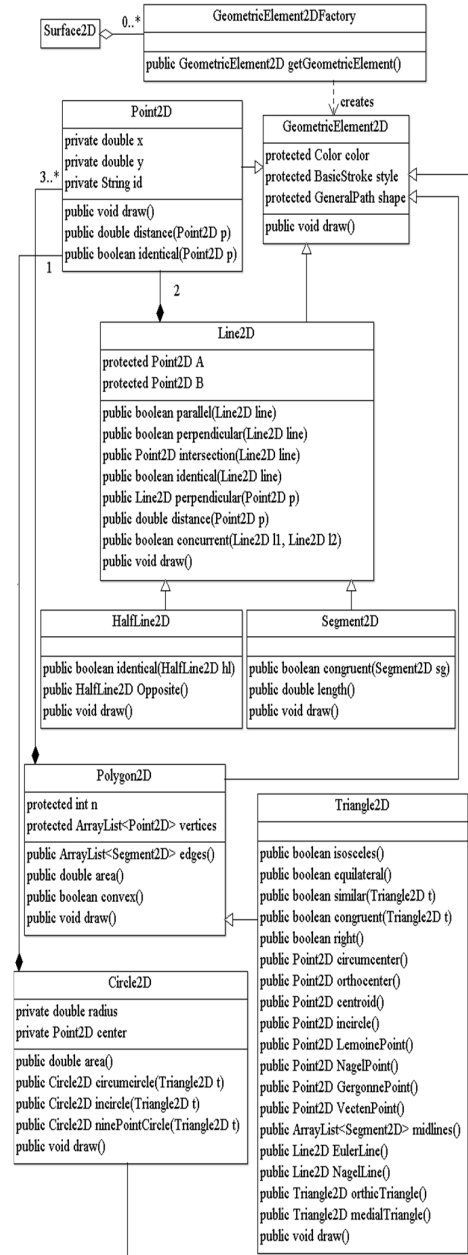


Fig. 2. Factory design pattern

Therefore, whenever an object of “Surface2D” type claims to get data from “GeometricElement2D” class, it is necessary to require firstly from the factory type. Factory template hides implementation details of an object by “GeometricElement2D” type from object of “Surface2D” type. Consequently, factory template is an elaborate pattern and very used in objectoriented programming concepts [14] for the computer application development.

4. SPECIAL POINTS IN A TRIANGLE

One of the software options allows the drawing of the most important points specific to a triangle. To determine the centroid [15] of a triangle, it were instantiated 11 objects which interact as can be seen in figure 3. Amongst the eleven objects that collaborate to determine the coordinates of the centroid four objects are of “Point2D” type, one object is of “Surface2D” type, one object is of “Triangle2D” type, three objects of “Segment2D” type and two objects of “Line2D” type. In figure 4 the centroid (denoted G) is represented along with three other points specific to a triangle: orthocenter (denoted H), incenter (denoted I) and circumcenter (denoted O).

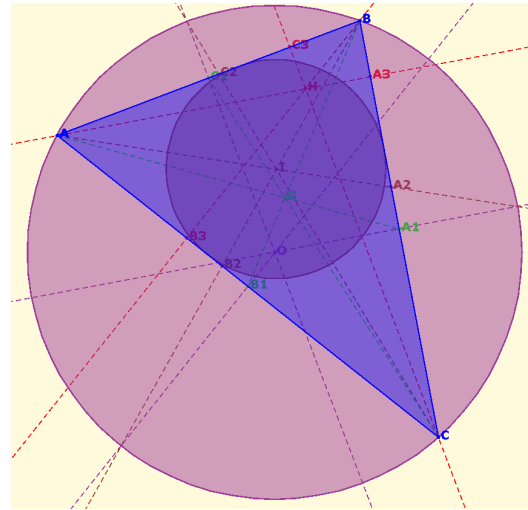


Fig. 4. Four important points in a triangle

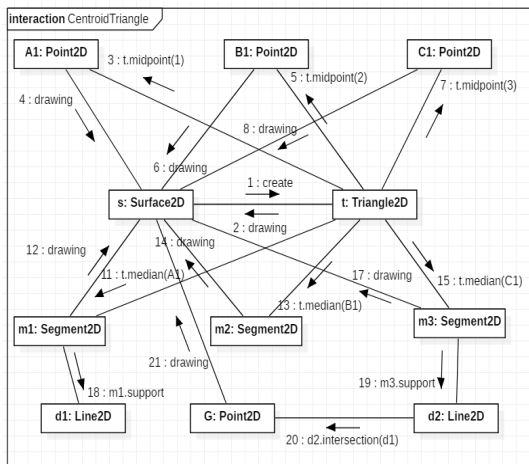


Fig. 3. UML collaboration diagram for drawing of the centroid

Another software option allows Lemoine point [16] representation as shown in figure 5. For the instantiation of an object of “Point2D” type obtained from the intersection of the triangle symmedians, thirteen objects interact as can be seen in UML collaboration diagram shown in figure 6.

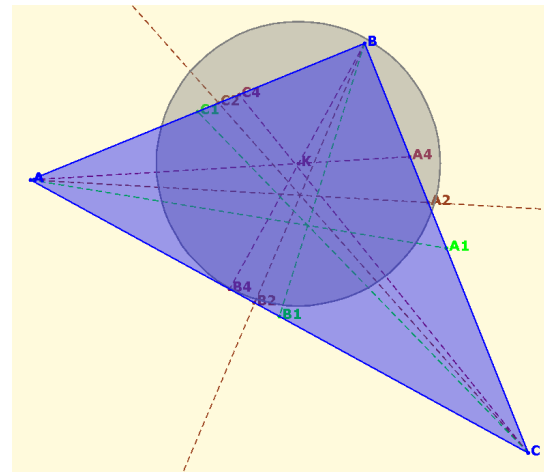


Fig. 5. Lemoine point

Amongst the thirteen objects that collaborate to determine the coordinates of the Lemoine point (denoted K) seven objects are of “Point2D” type, one object is of “Surface2D” type, one object is of “Triangle2D” type, two objects of “Segment2D” type and two objects of “HalfLine2D” type. This point is also known as Grebe point.

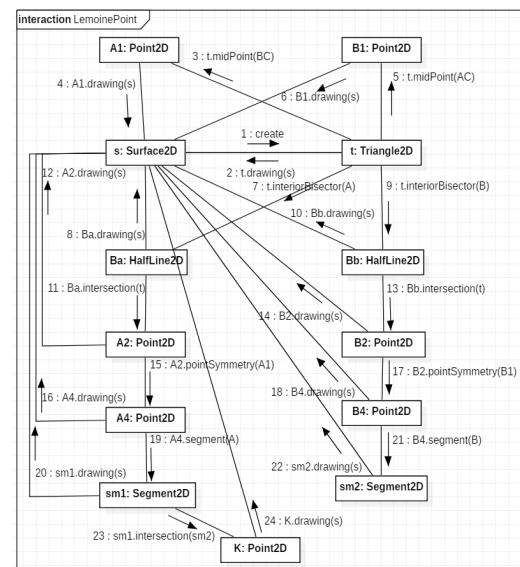


Fig. 6. UML collaboration diagram for drawing of the Lemoine point

The representation of Nagel point [15] is shown in Figure 7. For the instantiation of an object of “Point2D” type nine objects interact as can be seen in UML collaboration diagram shown in figure 8. Amongst the nine objects that collaborate to determine the coordinates of the Nagel point (denoted N) three objects are of “Point2D” type, one object is of “Surface2D” type, one object is of “Triangle2D” type, two objects of “Circle2D” type and two objects of “Line2D” type.

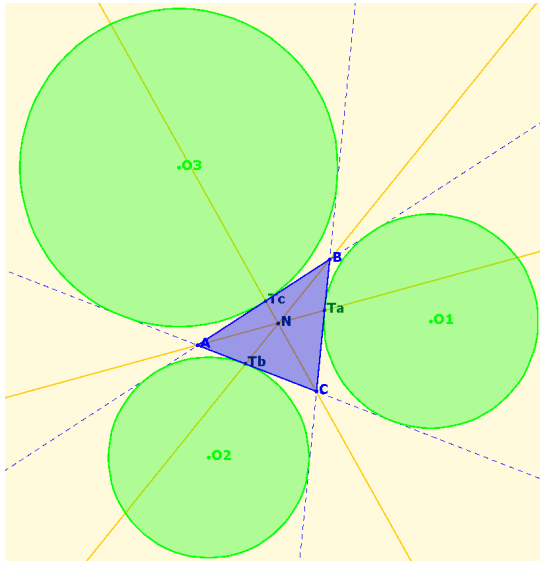


Fig. 7. Nagel point

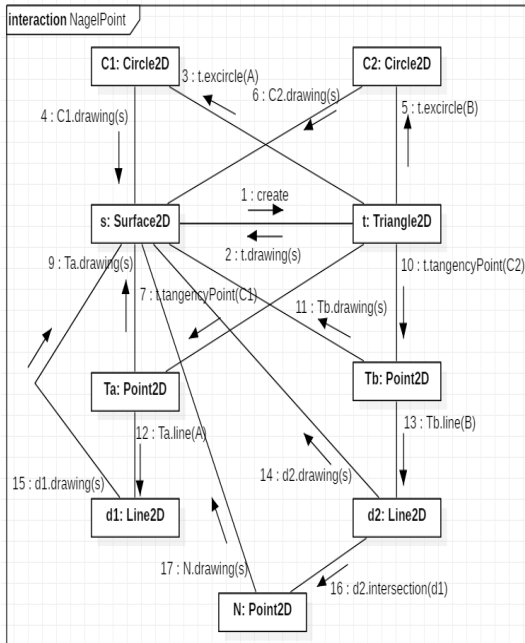


Fig. 8. UML collaboration diagram for drawing of the Nagel point

By selecting the option that allows to draw Gergonne point [16], a representation like the one shown in figure 9 is obtained. For the instantiation of an object of “Point2D” type eight objects interact as can be seen in UML collaboration diagram shown in figure 10. Amongst the eight objects that collaborate to determine the coordinates of the Gergonne point (denoted Ge) three objects are of “Point2D” type, one object is of “Surface2D” type, one object is of “Triangle2D” type, one object of “Circle2D” type and two objects of “Line2D” type.

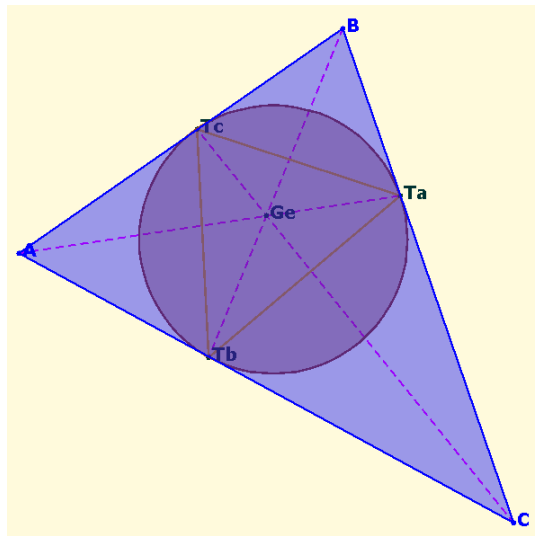


Fig. 9. Gergonne point

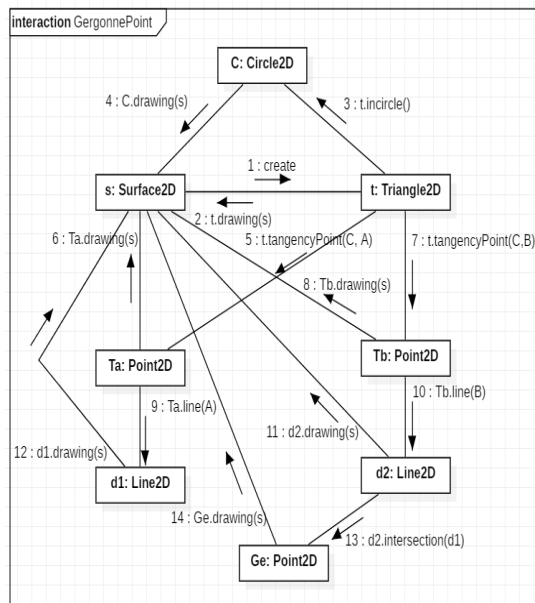


Fig. 10. UML collaboration diagram for drawing of the Gergonne point

By selecting the option that allows to draw outer Vecten point [15], a representation like the one shown in figure 11 is obtained. For the instantiation of an object of “Point2D” type nine objects interact as can be seen in UML collaboration diagram shown in figure 12. Amongst the nine objects that collaborate to determine the coordinates of the outer Vecten point (denoted V) three objects are of “Point2D” type, one object is of “Surface2D” type, one object is of “Triangle2D” type, two object of “Square2D” type and two objects of “Segment2D” type.

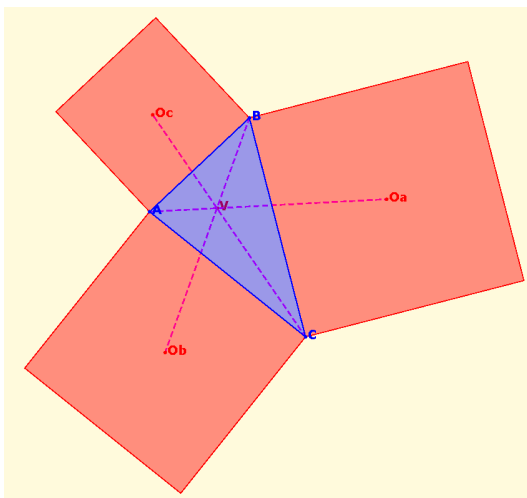


Fig. 11. Outer Vecten point

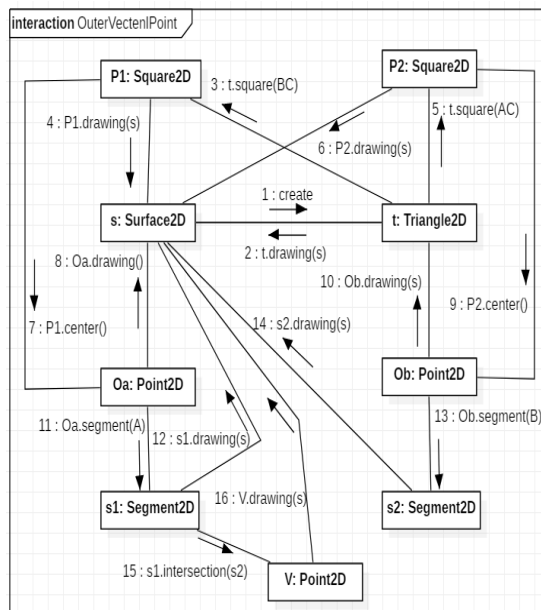


Fig. 12. UML collaboration diagram for drawing of the Vecten point

5. CONCLUSION

In this paper, the purpose was to design a triangle geometry computer application with and without using design templates. There were presented some issues that arise in the case of not using templates. This inefficient design has been optimized through creational template factory. To improve the software development, the factory template mandates the responsibility to the geometric element “Geometric-Element2DFactory”.

ACKNOWLEDGMENT

The paper presents results from the research activities of the project “Entrepreneurial competences and excellence research in doctoral and postdoctoral studies programs - ANTREDOC”, code: POCU/380/6/13/123927, co-financed by the European Social Fund through the Human Capital Operational Program 2014-2020, Priority Axis 6: Education and skills, “Support for post-doctoral students and researchers”. Beneficiary: Technical University of Cluj-Napoca / Partner: ROBERT BOSCH SRL.

8. REFERENCES

- [1] Gorton, I., *Essential Software Architecture*, Springer, Heidelberg, Germany, ISBN 978-3642191756, 2011
- [2] Pfleeger, S.L., Atlee, J., *Software Engineering: Theory and Practice*, Pearson, ISBN 978-0136061694, 2009
- [3] Fowler, M., *Patterns of Enterprise Application Architecture*, Addison-Wesley Professional, Boston, United States, ISBN 978-0321127426, 2002
- [4] Ghiormez, L., Panoiu, M., Panoiu, C., Rob, R., *Electric arc model in PSCAD — EMTDC as embedded component and the dependency of the desired active power*, IEEE 14th International Conference on Industrial Informatics, pp. 351-356, ISSN 2378-363X, Poitiers, France, July 2016
- [5] Morariu, R., Lupea, I., Anderson, C., *Noise analysis inside an electric vehicle during a run up on the road and on the chassis*

- dynamometer*, Acta Tehnica Napocensis, Series: Applied Mathematics and Mechanics, vol. 56, Issue III, ISSN 1221-5872, pp. 523-529, Cluj-Napoca, Romania, 2013
- [6] Cuntan, C.D., Baci, I., *Verifying the Functionality of a Parallel-Series Convertor using Nexis 4 Development Board*, Annals of Faculty Engineering Hunedoara, vol. XVI (4), ISSN 2601-2332, pp. 119-122, Hunedoara, Romania, 2018
- [7] Deteșan, O., *The dynamic modelling of the robot mechanical structure using the symbolic computation in Matlab*, Acta Tehnica Napocensis, Series: Applied Mathematics and Mechanics, vol. 56, Issue IV, ISSN 1221-5872, pp. 659-664, Cluj-Napoca, Romania, 2013
- [8] Timoftei, S., Brad, E., Sarb, A., Stan, O., *Open-source software in robotics*, Acta Tehnica Napocensis, Series: Applied Mathematics and Mechanics, vol. 61, Issue III, ISSN 1221-5872, pp. 519-626, Cluj-Napoca, Romania, 2018
- [9] Schonstein, C., *Kinematic control functions for a serial robot structure based on the time derivative jacobian matrix*, Acta Tehnica Napocensis, Series: Applied Mathematics and Mechanics, vol. 61, Issue II, ISSN 1221-5872, pp. 253-660, Cluj-Napoca, Romania, 2018
- [10] Deitel, H., *Java 9 for Programmers*, Pearson Education, USA, ISBN 978-0134777566, 2017
- [11] Altshiller, N., *College Geometry: An Introduction to the Modern Geometry of the Triangle and the Circle*, Dover Publications, USA, ISBN 978-0486458052, 2007
- [12] Gallatly, W., *Modern Geometry of the Triangle*, Forgotten Books, ISBN 978-1440088636, 2015
- [13] Sarcar, V., *Java Design Patterns: A Hands-On Experience with Real-World Examples*, Apress, Berkeley, CA, USA, ISBN 978-1484240779, 2018
- [14] Bruegge, B., Dutoit, A., *Object Oriented Software Engineering Using UML*, Prentice-Hall, Inc. Upper Saddle River, ISBN 978-0-13-606125-0, NJ, USA, 2013
- [15] Arbain, N.A., Azmi, M.S., Muda, A.K., Radzid, A.R., Tahir, A., *A Review of Triangle Geometry Features in Object Recognition*, IEEE 9th Symposium on Computer Applications & Industrial Electronics, pp. 254-258, ISBN 978-1-5386-8546-4, Malaysia, April 2019
- [16] Zhao, J., Liu, W., Xia, R., *A method of feature line extraction of triangle mesh surface model*, IEEE International Conference on Information and Automation, pp. 602-605, ISBN 978-1467322386, Shenyang, China, June 2012

Îmbunătățirea proiectării unui soft pentru geometria triunghiului utilizând un șablon creațional

Rezumat: Proiectarea unui sistem reprezintă, probabil, factorul cel mai critic ce poate afecta calitatea acestuia. Din fericire, în prezent există mai multe șabloane de proiectare a căror utilizare pot oferi o soluție de rezolvare a unei probleme generale. Deciziile de proiectare ale proiectanților experimentați sunt înregistrate sub formă de modele de proiectare. Astfel, fiecare șablon de proiectare se concentrează pe o anumită problemă de proiectare orientată obiect. În această lucrare este prezentat un soft interactiv care poate fi utilizat în studiul geometriei triunghiului. Proiectarea sistemului este îmbunătățită prin utilizarea șablonului creațional "Fabrica", iar implementarea este realizată prin intermediul limbajului de programare Java.

Anca-Elena IORDAN, PhD, Lecturer, Polytechnic University Timisoara, Department of Electrical Engineering and Industrial Informatics, anca.iordan@fih.upt.ro, Revoluției 5 Street, Hunedoara, 331128, ROMANIA, +40724578986

Florin COVACIU, PhD Eng., Lecturer, Technical University of Cluj-Napoca, Department of Design Engineering and Robotics, florin.covaciu@muri.utcluj.ro, Blvd. Muncii 103-105, Cluj-Napoca, 400641, ROMANIA, +40755566491