

1.3. Філософські проблеми науки

Bondar S.

bulkadog@bigmir.net

PhD, Associate Professor

Department of Philosophy

Kyiv National

Economic University named by Vadym Hetman

Kyiv, Ukraine

PROGRAMMING PARADIGMS:

THE NECESSITY OF PHILOSOPHICAL ANALYSIS

Programming is of particular interest to philosophical analysis. It unites the anthropocentrism inherent in social sciences with the physicalism of the natural sciences. The strength and weakness of cognition is manifested in it in the clearest form. Programming is one of the technologies that determine the future of civilization and it is also the application scope of the achievements of the basic and applied sciences. At the same time, programming is carried out to obtain a particular practical result, the programmer is rarely interested in the theoretical side of the issue, as a consequence, this result may become unexpected even for the creators themselves.

Significant practical achievements in software engineering, obtained in recent decades, require a theoretical generalization and justification of the selection of a particular software methodology and its technical implementation. Theoretical reinforcement is designed to eliminate the unnecessary complication of tools and technologies, to increase the reliability and efficiency of programming languages. It helps to bridge the gap between the level of abstraction of programming languages and understanding the basic principles of their organization and functioning, methods of software development.

Gorodnyaya L.V. defines programming paradigms as “alternative approaches to information processing, established during the creation and application of programming languages and systems” [1]. Programming paradigms are determined not only by what we are able to conceptualize but also by what we can technically realize from the

conceptualized. L. Jerinic writes: “a language that supports a paradigm well is often hard to distinguish from the paradigm itself in practice” [2]. The programming paradigm is a methodological principle for constructing a programming language and its functioning. Programming languages model the informational characteristics of the real-world, form plausible models of objects and processes of reality, which can be considered as the technical implementation of the set of knowledge, norms, and values. The paradigm becomes the basis for constructing such models. Language is an abstraction of a physical machine, and the paradigm defines the principles of abstraction and the essential elements that should be highlighted to enhance the expressiveness of a language. The choice of a paradigm is difficult because it is necessary to take into account the type of data and the operating conditions of the software product.

In the process of cognition, we always deal with the “pre-established harmony” of human nature, with its inherent uniqueness, which is taken for universality. Programming languages reproduce the anthropocentric view of the world and fears regarding the alienness and danger of a software product created on their basis, in particular artificial intelligence, indicates an incomplete understanding of human nature. Nietzsche’s thesis: «To the extent that human beings are understood at all these days, they are understood like machines» [3, 12] is still relevant.

The paradigm determines the functioning of the language and provides its formal description. On the basis of the principles introduced by it, incorrect propositions are eliminated and the correct ones are generated. The paradigm is a set of principles that generate a program. It not only sets the principles of the device and the functioning of the program; it is the process of its creation.

Programming languages are evolving in the direction of increasing accuracy and flexibility, freedom from technical problems and limitations of the physical medium. One of the driving forces of this evolution is the mechanism of abstraction and formalization. Formalization, from the point of view of Fedotov A.M., expands the range of tasks for the solution of which they can be involved [4]. The author writes about the multiparadigmality of long-lived programming languages and the prospect of creating a unite programming

paradigm that combines the most reliable and proven methods of producing an information product. Creating such a paradigm is problematic due to the diversity of tasks solved by modern programming, but most importantly, the number of technical limitations.

What is a “good” programming paradigm? It allows performing a large number of calculations using a compact program. A language, supported by the paradigm, should be simple and easy to learn, similar to natural languages and contain powerful operators. It should have conceptual integrity, and, at the same time, the possibility of easy updating of its components. Equally important is good data structuring.

According to Demyanakov V.Z. “Belonging to a paradigm is not only manifested in research methods but also signaled by the common metalanguage and the style of the presentation of thoughts” [5]. Unlike programming techniques and styles, paradigms are multidisciplinary. The programming paradigm is a way to solve technical challenges in the field of programming, implementing a project taking into account the theoretical base, technical capabilities, and socio-cultural conditions.

References:

1. Gorodnyaya L.V O klassifikatsiyah paradigmi programirovaniya i parallelnom programirovanii. Available at: < <https://cyberleninka.ru/article/n/o-klassifikatsiyah-paradigmi-programirovaniya-i-parallelnom-programirovanii/viewer> > [Accessed 01.02.2020]

2. Jerinic L. A Perspective on Combining Different Programming Paradigms. Available at: <https://www.academia.edu/3142608/A_Perspective_on_Combining_Different_Programming_Paradigms > [Accessed 26.02.2020]

3. Nietzsche F. The Anti-Christ, Esse Homo, Twilight of the Idols. URL: http://krishnamurti.abundanthope.org/index_htm_files/The-Anti-Christ-Ecce-Homo-Twilight-Of-The-Idols-And-Other-Writings-by-Nietzsche.pdf [Accessed 26.02.2020]

4. Fedotov A.M. Paradoksy informacionnyh tehnologij. Available at: <<https://cyberleninka.ru/article/n/paradoksy-informatsionnyh-tehnologiy/viewer> > [Accessed 26.02.2020].

5. Dem'jankov V. Paradigma v lingvistike i teorii yazyka. Available at: <<http://www.infolex.ru/FORKB09.HTM>> [Accessed 26.02.2020].

Макарова А.О.

Makarova_AO@ukr.net

кандидат філософських наук, доцент,
докторант кафедри філософії та політології
Житомирського державного університету імені Івана Франка
м. Житомир, Україна

АРХЕТИП ЯК ЕЛЕМЕНТ МІЖДИСЦИПЛІНАРНОГО ДИСКУРСУ СОЦІАЛЬНОЇ ДІЯЛЬНОСТІ

XX століття у філософії було часом дискурсу і наративу, часом уваги не до цифри, а до символічного знаку, часом не проектування, а спілкування. Багато дослідників погоджуються на тому, що ключовим моментом у цій спільній налаштованості на діалог був антропоцентризм [1, 10-13]. Здавалось би, центрація (концентрація) людини на людині – найприродніший з усіх способів ставлення, бо лише людина є мірою добра і зла, отже, потрібно просто правильно провести межі і розставити акценти. В цьому мало допомогти право як норма та філософія як думка. XX століття – час людського Logos – таким чином постійно нагадувало людині, що вона істота, здатна до розумного культурного і міжкультурного діалогу, заснованого на дотриманні меж. І хоча самі межі постійно змінювалися, відсувалися, переглядалися, основні межі все ж лишалися незмінними – любов, милосердя, толерантність, ерудованість, інтелігентність, відкритість, порозуміння. Сама поява у науковому дискурсі соціології на тлі безапеляційного тріумфу структурного функціоналізму феноменологічної, символічно-інтеракціоністської методологій вказує на те, що у XX столітті суб'єкт взяв від соціальних моделей «своє», відвоював належне йому місце живої, емоційної, непрогнозованої духовної сутності.

Спокусливо-легко звідси перейти до виявлення виключно позитивів «множинних реальностей» як «відкритих світів» [2, 137]. І все ж у XXI столітті те, що раніше було цінним та ціллю – людяність – трансформується у проєктивний