

OBJECT-ORIENTED PROGRAMMING TEACHING IN ENGLISH THROUGH COMPUTATIONAL FAIRY TALES TO NON-NATIVE SPEAKERS

Lesia V. Pikun

Foreign Language Department Chernihiv Polytechnic National University, Ukraine

ORCID iD: Lesia V. Pikun

 <https://orcid.org/0000-0002-3072-2173>

Abstract. *In modern programming education, the object-oriented approach is one of widely used methodologies and a basic component of university curricula. However, despite all its advantages, object-oriented programming is difficult to learn and requires significant time and effort to understand its concepts and use them effectively. Higher education institutions play a crucial role in helping students develop strong object-oriented programming skills, especially in English. One of the possible solutions to these challenges is the use of Computational Fairy Tales by Jeremy Kubica. Written by a native speaker and computer science expert, these stories simplify complex programming principles by supporting technical English vocabulary and cultural context. J.Kubica's Computational Fairy tales integrate object-oriented programming concepts with professional English language learning. This engaging and effective approach helps non-native speakers master programming concepts while improving their communication skills for IT careers. With the author's permission, these texts have been adapted into methodological guidelines designed to enhance reading comprehension and professional communication in object-oriented programming. The adapted materials include pre-reading, reading, and post-reading activities aimed at developing vocabulary, comprehension, and analytical skills.*

Key words: *computational fairy tales, educational materials, IT education, methodological guidelines, object-oriented programming, professional communication.*

1. INTRODUCTION

Object-oriented programming is one of the recently developed methodologies in software that is becoming popular. It supports software engineering giving effective methods for organisation and reusability (Anderson, 1988). Object-oriented programming is a methodology that covers the design, writing, testing, and maintenance processes of computer programs. It is a key methodology for training specialists in computer technology (Rangiseti, 2024). Object-oriented programming is important for making user-friendly software. It helps transform complex concepts into handy software components, like classes while encapsulating details to simplify interactions. This approach improves system flexibility by controlling data access and facilitating easy software updates. Key principles of object-

Submitted January 31st, 2025, accepted for publication March 7th, 2025

Corresponding author: Lesia Pikun, Foreign Language Department Chernihiv Polytechnic National University, Ukraine

E-mail: lesyapikun@stu.sn.ua

oriented programming like inheritance and polymorphism play an important role in creating reusable and adaptable software solutions (Rangisetti, 2024). The knowledge of object-oriented programming concepts and principles, along with mastering the tools of this methodology makes it possible to create well-organized and effective programs. Object-oriented programming has changed the way code is written (Koundilya, 2021). Engineers use objects, classes, and the structural data of object-oriented programming to create productive and reliable applications in Java, C++, and Python programming languages. They develop software tools to solve cybersecurity problems. Currently, object-oriented programming is a leading approach in application programming and serves as an effective tool for creating complex commercial software systems (Object-oriented programming in Java, 2019, p.10).

It is important to study object-oriented programming in English for some reasons. Object-oriented programming terminology in English is widely accepted worldwide. Learning these concepts in a different language can lead to difficulties when you encounter the original English terms in code, documentation, or in discussions with other programmers. Furthermore, English is mostly widely used language in the tech industry. Most programming languages, frameworks, documentation, tutorials, and codebases are written in English. Consequently, learning object-oriented programming concepts and terminology in English allows you to understand and use the existing resources.

2. THEORETICAL FOUNDATIONS

In contemporary programming education, the object-oriented approach is one of the commonly used practices. That is why it has become an essential component of university curricula. Object-oriented programming has many advantages; however, it is harder to learn than procedural programming. Mastering its concepts and applying them in practice requires time and effort. D.Saidova, a teacher in the Department of Algorithms and Programming at Karshi State University, highlights the challenge high educational institutions face in preparing future computer science educators. To teach object-oriented programming effectively, these educators must develop both subject knowledge and strong teaching strategies. In the article, "Analysis of the Problems of the Teaching Object-Oriented Programming to Students" D. Saidova observes the challenges associated with teaching object-oriented programming to students (Saidova, 2022). One possible solution to this task is the usage of "Computational fairy tales" by Jeremy Kubica. An expert in IT technologies and a native speaker writes these texts in English. The fairy tales provide material for IT professionally oriented English language learning. Jeremy Kubica's "Computational Fairy Tales" (2011-2012) explain complex object-oriented programming concepts in a simple and engaging way. These stories also help students strengthen their English language skills in professional sphere. Authentic materials like these help students to learn technical vocabulary in real-life context, understand nuances, and practice the reading comprehension skills necessary for interpreting technical texts.

3. ANALYSIS OF RESEARCH AND PUBLICATIONS

Teaching object-oriented programming is an important aspect of higher education. The authors of the article "Ranking of Problems and Solutions in the Teaching and Learning of Object-Oriented Programming" discuss the challenges and strategies involved in

teaching this subject. They emphasise that programming skills are important for professionals in system development and highlight the need for programming courses to balance theoretical knowledge with practical application. Traditional teaching methods in programming are often criticised for being ineffective (Gutiérrez, 2022). The article “Development of Project Based Learning Model in Object-Oriented Programming Learning in Polytechnic” seeks to enhance teaching methods for programming and improve student outcomes in object-oriented programming (Idris, 2024).

There are language barriers for students who do not speak English as their first language. Learning object-oriented programming becomes even more complicated for non-native speakers. They must simultaneously master technical concepts and new terminology in a foreign language. Ukrainian specialists in vocationally oriented education aim to develop students’ communicative competence for professional use. This includes encouraging independent learning in foreign language (Chernichenko, 2010). Motivation is a key factor in language learning. Vocationally oriented education integrates foreign language learning with professional subjects to enhance knowledge acquisition (Barabanova, 2005). Authentic texts play a crucial role in this process by expanding students’ professional vocabulary, making learning more engaging and relevant, and encouraging independent knowledge acquisition. Teachers should carefully select educational materials that align with students’ professional fields, keeping them engaged through relevant topics and real-world applications (Ketsyk-Zunchenko, 2019). To achieve these goals, modern IT tools should be incorporated into teaching, allowing students to practice real-world professional tasks. This approach enhances critical thinking, debate skills, and overall training for their future careers (Cherednichenko, 2018).

The selection of career-oriented material for non-linguistic specialties is crucial. This material should be relevant, informative, and supporting the discipline, developing speaking and writing skills on professional issues. Authentic sources such as specialised literature and periodicals, along with Internet resources, ensure up-to-date content. Texts should also include exercises for vocabulary building and practical language use, helping students apply their foreign language skills in real-world professional settings (Shevchuk, 2023).

English for Special Purposes plays a vital role in professional education. I.F. Tserbuk, N.V. Kosylo, and O.V. Semotiuk argue that studying career-related materials in a foreign language fosters students’ motivation to acquire specialised knowledge and improve language skills (Tserbuk, 2012). However, practical application is crucial. Without real-life examples and tasks, students may struggle to understand complex concepts. Therefore, teachers must provide educational materials in English that support hands-on learning.

In Ukraine, significant attention is given to selecting professionally oriented texts for IT students. These texts should align with curriculum topics, match students’ interests and career needs, and stimulate motivation and engagement in learning. According to E. Roshchyna, professionally oriented texts bridge the gap between acquiring specialized knowledge and mastering a foreign language. They also encourage analytical thinking, scientific reasoning, and creativity (Roshchyna, p.4).

Career-oriented texts are the basis of the content of teaching a foreign language for professional purposes, one of the main sources of teaching professional vocabulary in context. The content of these texts directs effective learning activities, creates situations for searching and analysing certain professional information. Well-chosen texts also help to develop students’ analytical thinking and scientific reasoning, encouraging them to be active and creative. The topics of the texts should inspire students to further dialogue and

express their own opinions, making professionally oriented texts a means of intellectual, social, and professional development (Tsebruk, 2012).

According to H. Barabanova, the main tasks in learning a foreign language through professionally oriented literature include acquiring information that matches and expands the material learned in specialised subjects and developing skills for independent reading of professional literature (Barabanova, p.76). H. Cherednychenko and co-authors suggest that reading skills can be developed through working with original professional texts, including extensive reading with comprehension examination. The scholars highlight three main stages of work: pre-text, text, and post-text, each with specific tasks aimed at developing professional and communicative skills (Cherednychenko p.152).

4. OBJECTIVE

The purpose of this article is to highlight the importance of integrating object-oriented programming into English language teaching for professional purposes. The goal is to help students not only gain a deep understanding of the key programming concepts but also develop professional communication skills in English that correspond with modern labour market demands. The article explores effective methods, including the use of creative learning materials, such as Jeremy Kubica's fairy tales.

5. PRESENTATION OF THE MAIN MATERIAL

The development of object-oriented programming has been a dynamic process. It started in the 1960s and 1970s with such object-oriented programming languages as Simula and Smalltalk, which laid the foundation for key concepts such as classes, objects, and inheritance. In the 1980s, C++ combined procedural and object-oriented programming so object-oriented programming became more popular. In the 1990s, Java made object-oriented programming more widespread with its ability to run on any platform. In the 21st century, languages like Python, Kotlin, and Swift added more flexibility, improving performance and adaptability. Object-oriented programming's evolution continues nowadays to meet the demands for better performance and easy usage (Koundilya, 2021).

Object-oriented programming plays a crucial role in training future professionals in computer technology. It is a basis for many modern software products making them more structured, integrated, and easy to maintain. At the Chernihiv Polytechnic National University, object-oriented programming is taught as a separate course during the second year in the Software Engineering Programme and Computer Engineering bachelor degree Programme.

There are several key advantages of teaching and learning object-oriented programming and English language at the same time. First, English is the primary language of international communication in the IT sector. Second, being proficient in the English language opens opportunities for professional mobility, improving graduates' competitiveness in the job market. Third, many educational materials, including books, articles, and online courses on object-oriented programming, are available only in English. Understanding English allows students to use these materials directly, leading to a more profound grasp of the subject. In addition, learning English in the context of object-oriented programming helps to develop the communication skills essential for effective teamwork, participation in conferences, and writing scientific papers.

There are occasions when scientists have used fiction to share scientific ideas with a wide audience and to promote sciences and computer technologies. Among such scientists-writers, there are visionaries, communicators, and educators. Isaac Asimov (1920-1992) is a visionary scientist. He was a professor of biochemistry at Boston University, who predicted the advent of video communication, which became a reality with the advent of technologies such as Skype and FaceTime, automation of kitchen appliances and robotics, and computer miniaturisation. Isaac Asimov also wrote popular science books explaining the natural sciences (e.g. *The Intelligent Man's Guide to Science*, 1960 and *The Universe. From Flat Earth to Quasar*, 1966). Vernor Vinge (1944-2024) was an American mathematician, programmer, and writer who was one of the pioneers who described cyberspace. He introduced the concept of the technological singularity. Vernor Vinge researched the topic of artificial intelligence, the unforeseen impact of the Internet as well as on society and artificial intelligence. Greg Egan (born 1961) is an Australian writer with a degree in computer science. He explores such compound scientific ideas as quantum mechanics, artificial intelligence, simulated reality, and consciousness uploading. Another well-known communicator was an American astronomer and planetary scientist with Ukrainian roots Carl Sagan (1934-1996). The works of these authors reflect a profound understanding of science and technology and their ability to see future trends and their potential impact on the future of humanity. Most of these authors have used their knowledge to write captivating stories.

There are also those writers who have used scientific concepts to write stories for didactic purposes. The name of the mathematician Charles Lutwidge Dodgson (1832-1898), better known by his literary pseudonym Lewis Carroll, can be mentioned. His scientific studies inspired him to write fairy tales. As a professor of mathematics, he included logical problems and paradoxes in the fairy tale "Alice's Adventures in Wonderland", and the idea of mirror images and symmetry in "Through the Looking-Glass".

Jeremy Kubica is a modern storyteller in the field of computer science. He is a specialist in data mining, searching for structure in large data sets. Jeremy Kubica has a bachelor's degree in computer science from Cornell University and a master's degree in robotics from Carnegie Mellon University. He wrote his thesis on large-scale problems of data mining and structure search in large data sets with noise. Jeremy Kubica has experience working at Google, where he led the development and application of artificial intelligence, machine learning, and data science in the Cloud AI organisation. He was the head of advertising quality improvement in the Pittsburgh office, including online quality prediction, advertiser policy validation, distributed machine learning infrastructure, and statistical analysis. Currently, Jeremy Kubica is the project development director for the LINCC Frameworks (LSST Interdisciplinary Network for Collaboration and Computation), where he leads a team working at the intersection of modern computer science and astrophysics.

Jeremy Kubica has written a number of computer fairy tales. The scientist used the fairy tale genre to explain complex concepts of object-oriented programming in a simple and engaging way. Jeremy Kubica's computer fairy tales on object-oriented programming are good material for teaching English for professional purposes to students studying computer science, information systems and technologies, computer engineering, software engineering, systems analysis, and cybersecurity. These tales provide a good basis for conveying complex programming concepts in an accessible and understandable way. Fairy tales by Jeremy Kubica are the source that contributes to the acquisition of professionally oriented knowledge and the development of skills and abilities to communicate in English, as they are written by a native speaker and a specialist in the field of computer

science (Kubica, 2011-2012). Firstly, the texts have a professional orientation and include concepts, terms and facts that are useful to students in their professional activities. Secondly, the lexical material of the texts corresponds to the B2 and C1 levels. Thirdly, fairy tales are characterised by novelty and relevance of information and this approach encourages students to read these texts. Fourthly, the fairy tales are authentic and appear to be the product of a native speaker's speech activity, without adaptation. These fairy tales create and demonstrate the real use of various language constructions and bring students closer to understanding the foreign language culture. The usage of authentic texts in the professional subject gives students the opportunity to compare information with the material studied in the core subjects in their native language. In this way, the principle of professional orientation of teaching disciplines is implemented. Fifthly, the text of every fairy tale is connected on the previous text and repeats and develops the previously studied material in order to form consolidation and improvement.

The series of fairy tales devoted to object-oriented programming include five episodes. The first fairy tale, "Data Validation, Marcus, and the Cheese Minstrel: Part 1 of Marcus and the Cursed Cheese", addresses the problem of data validation and its implementation, which is important for ensuring the safety and reliability of software. In this fairy tale, food validation is used as a metaphor for data validation. Marcus, a wizard, meets the Minstrel of Cheese, who is sitting with a large block of cheddar. When Marcus asks for directions to the Castle of Cheswick, the minstrel reveals that the Castle of Cheswick is actually a cheese factory. Curious about Marcus's mission, the minstrel learns that Marcus received cursed cheese and needs to investigate. Fascinated by Marcus's explanation of food validation, the minstrel gives directions to the castle and asks to join Marcus on his journey (Kubica, 2011). Together they travel to the Castle of Cheswick. According to the plot of this computational fairy tale the author draws parallels between Marcus's actions and the process of data validation. Just as the wizard checks his food to ensure it meets his expectations, data validation ensures that data is correct and meets requirements. The story by Jeremy Kubica emphasises the importance of validation process before using it, just as data should be validated before being processed or used in applications. This includes examination for accuracy, completeness, and consistency. The fairy tale highlights the potential problems that can arise from not validating inputs, such as receiving cursed cheese or having a pizza fall apart. Similarly, not validating data can lead to errors, security vulnerabilities, and system failures. Overall, the fairy tale illustrates the importance of thoroughly checking and validating inputs to prevent issues and ensure reliability, whether it is food or data.

The second story is called "Objects, Encapsulation, and Cheese Factories: Part 2 of Marcus and the Cursed Cheese". In the fairy tale Marcus argues with the foreman of the Castle of Cheswick Cheese Factory, insisting his cheese was cursed. Marcus explains that the curse targets wizards and questions the machines' reliability. The foreman admits they don't test their cheese for curses. Marcus learns about a mysterious visitor who inspected a machine and the shipping facility. Marcus decides to investigate the shipping area (Kubica, 2011). The fairy tale uses the cheese-making machines as a metaphor to explain data and methods for operating on the data. The machines represent data with various attributes like temperature, tray orientation, and humidity. These attributes represent the data that the machines work with. The machines have methods such as Bake, Cool, and SpinTray. These methods are operations that manipulate the attributes to produce the desired outcome, which in this case is high-quality cheese. The machines

hiding their internal machinery from the operators illustrate the concept of encapsulation. The operators interact with the machines through simple methods without a need to understand the complex internal processes. The foreman mentions the usage of unit tests to ensure the machines are functioning correctly, which parallels the idea of validating data to ensure it meets the required standards before it is used in an application. Overall, the story explains how data and methods work together within a system, and how encapsulation helps manage complexity by hiding internal details and exposing only necessary functionalities.

The third fairy tale highlights the concept of object class. The tale is called 'Class of Cheese: Part 3 of Marcus and the Cursed Cheese' and uses the example of different types of cheese to explain this concept, as well as individual instances of the class. Marcus and the cheese minstrel visit a room filled with various cheeses. The foreman explains they are packing a shipment of soft cheeses and preparing Swiss cheese blocks, which are made by machines using the same recipe but with different attributes (Kubica, 2011). The fairy tale uses the concept of cheese blocks to explain how a class defines the type of an object in object-oriented programming. The foreman describes the Swiss cheese blocks as being created by machines using the same recipe, which represents a class in object-oriented programming. A class serves as a blueprint for creating objects, with defined attributes like weight, size, and number of holes, and behaviours like emitting smell. In object-oriented programming, a class defines the attributes (data) and methods (functions) that its objects will have. The foreman mentions that the cheese blocks are different in attributes of the Swiss cheese class. In object-oriented programming, objects are examples of a class, each with its individual specific values for the attributes defined by the class. Despite having different values for their attributes, all Swiss cheese blocks share the same set of attributes and behaviours defined by their class. This illustrates how a class ensures consistency across its objects in object-oriented programming. By using the cheese-making process as a metaphor, the fairy tale effectively explains how a class in object-oriented programming defines the type and structure of its objects.

The final tale in the series about the cursed cheese is "Inheritance in Cheeses and Magic Spells: Part 4 of Marcus and the Cursed Cheese". The foreman recalls that the visitor was particularly concerned about the type of cheese being shipped. Marcus explains that the wizard who cursed his cheese must have been highly skilled, as he cursed it without knowing its type. This is significant because different types of cheese have unique properties. Marcus then learns that the visitor's arrival coincided with Sam, the logistics manager, becoming obsessed with solving a complex routing problem. Marcus identifies this as a powerful curse, which causes him to lose his appetite for cheese and tells that he can break it. Marcus also understands why he was targeted and considers this as a broader danger to the whole kingdom (Kubica, 2011). This tale reveals such an aspect of object-oriented programming as inheritance, which means the ability to create derived classes (or subclasses) of a class. These derived classes can reuse attributes or code defined in the original (or base) class. Marcus explains that all classes of cheese are derived from a common "Cheese" base class. This base class includes general properties like being made from milk, having weight, and density. Each specific type of cheese, such as bleu cheese or Swiss cheese, is a subclass of the base "Cheese" class. These subclasses inherit the general properties of the base class but also have their own unique properties and behaviours. For example, some cheeses might have a popping sensation when eaten. The story illustrates that subclasses inherit attributes and methods from the base class. This means that while all cheeses share common characteristics (inherited from the base class), each type of cheese can also have additional properties and behaviours specific

to that type. By using cheese as a metaphor, the fairy tale effectively explains how inheritance allows for the creation of specialized subclasses that build upon the properties and methods of a base class in object-oriented programming.

In object-oriented programming, inheritance is the mechanism of creating new classes based on existing class. The new classes are subclasses. They receive the properties and methods defined in the original, or base, class. Besides inheriting attributes and methods from their base class, subclasses can include their own unique code. This process is described in the fairy tale ‘Classes, Inheritance, and the Three Little Pigs’. It is a kind of adaptation of the well-known folk tale “The Three Little Pigs”. Three little pigs are knowledgeable in architecture and object-oriented programming. They build houses using different subclasses of a common House class. The youngest builds a “StrawHouse”, the middle a “WoodHouse”, and the oldest a “SecureBrickHouse”. A big bad wolf arrives and tries to blow down the houses. The first two pigs escape to their older brother’s “SecureBrickHouse”, which withstands the wolf’s huffing and puffing. Safe inside, the pigs discuss the merits of their house designs and consider creating a new “SecureWoodHouse”. Meanwhile, the wolf exhausts himself trying to blow down the brick house (Kubica, 2012).

With the author’s permission, fairy tales by Jeremy Kubica were used as a basis for creating methodological guidelines designed to develop reading skills as well as oral and written professional communication on the topic of object-oriented programming. The fairy tale texts serve as an educational material for developing different IT and English language skills.

The texts by Jeremy Kubica are particularly valuable in teaching reading. Proficiency in reading is achieved through a sequence of pre-text, text, and post-text activities (Extra Reading Practice, 2024). Pre-reading stage helps students to remove difficulties in understanding the content of the text and its characters. For example, the perception of a minstrel, a foreman, a logistics manager is important to understand the function of these characters in the fairy tale. The knowledge of the types of cheese is an important part the author uses to illustrate the category of classes and subclasses in object-oriented programming. The Ukrainian students might be unaware of “The Three Little Pigs” iconic phrase “I will huff and puff and blow your house in”. Pre-reading stage also provides a revision of reading rules, which is crucial, especially for students with varied English proficiency. It helps to establish foundation for text understanding, builds confidence for lower-level learners, who might struggle with unfamiliar words, and improves engagement, as the students can accurately read words.

Post-reading activities focus on prepositions, synonyms, idioms, and sentence patterns. These exercises extend vocabulary knowledge because synonym practice helps students enlarge their vocabulary. Working with prepositions and sentence patterns allows students to understand common English structures and relationships between words better, because they are often difficult for learners. Idioms and varied sentence patterns help students understand natural, conversational English and support their comprehension of authentic materials, such as books and movies. Post-reading tasks encourage students to engage the text beyond comprehension, promoting an active learning process where they apply new language forms. Post-reading questions help ensure that students understand the key points and main idea of the text, allowing them to reflect on what they have read. The questions about the concepts of object-oriented programming encourage students to go beyond the superficial understanding by promoting them to analyse, interpret, and evaluate the text. Answering the questions,

describing the illustrations and a list of events allows students to practice making complete sentences, using vocabulary from the text, and structuring their thoughts.

Post-reading activities include the tasks for creating dialogues and monologues to illustrate idioms, set phrases and proverbs. Practising idioms and proverbs through dialogues or monologues help students learn how to use these expressions naturally in real-life conversations, making their language use more authentic and fluent.

6. CONCLUSIONS

Object-oriented programming is a widely adopted methodology in software development that significantly improves software engineering by promoting organisation and reusability. It plays a crucial role in designing, writing, testing, and maintaining programs, making it essential for building flexible, scalable, and user-friendly software. However, object-oriented programming can be more difficult to learn than procedural programming, requiring substantial time and effort to master. Educators often face difficulties in teaching object-oriented programming concepts, but resources like "Computational Fairy Tales" by Jeremy Kubica, which simplify technical ideas while also teaching English language skills, provide helpful solution. Fairy tales by Jeremy Kubica are an excellent educational resource, as they not only explain complex programming concepts in an understandable way, but also serve as a powerful tool for learning vocabulary and enhancing reading and speaking skills. The lexical material in these fairy tales helps improve foreign language proficiency, while additional exercises support the development of communicative competencies necessary for professional programming tasks. Reading J. Kubica's fairy tales, students not only enhance their communicative reading skills but also develop intellectual and communicative reading abilities. The dialogues between characters in the fairy tales further teach language competence, enabling students to communicate professionally in various scenarios where such communication is required.

J. Kubica's fairy tales provide a solid foundation for creating methodological guidelines to develop reading skills and professional communication within the context of object-oriented programming. The texts support learning through a structured approach that includes pre-reading, reading, and post-reading activities. These stages help students overcome comprehension difficulties, expand their vocabulary, and engage with complex topics. Post-reading tasks, including exercises on prepositions, synonyms, idioms, and sentence patterns, further enhance language proficiency while encouraging deeper analysis of the text. By using these fairy tales, students not only strengthen their understanding of object-oriented programming concepts but also improve their ability to communicate effectively in English, making the learning process both engaging and practical.

REFERENCES

- Anderson, B. Object-oriented programming. *Microprocessors and Microsystems*. Vol. 12, Issue 8, pp. 433-442. [https://doi.org/10.1016/0141-9331\(88\)90135-4](https://doi.org/10.1016/0141-9331(88)90135-4) (1988)
- Barabanova H.V. *Methods of Teaching Professionally-oriented reading in a Non -Language University*, Kyiv. 315 p. (2005).
- Cherenichenko H. Modern Trends in Professionally Oriented Foreign Language Teaching of Technological Specialities Students. *Scientific Practices of Higher Educational Establishment "Donetsk National Technical University"*. Series: Pedagogy, Psychology and Sociology № 2 (23) <https://pedagogy.donntu.edu.ua/2-25-2018/h-a-cherednichenko/> (2018)

- Cheredichenko H.A., Shapran L.Yu., Kynytsia L.I. Methods of teaching foreign language professionally oriented reading to students of non-native language higher education institutions. *Teaching Languages in Higher Education at the Present Stage*. Issue 17. 150-157 (2010).
- Extra Reading Practice in Object-Oriented Programming. Methodical Instructions in English for Self-Study on Home Reading for Bachelor Students in the Specialities 121 Software Engineering and 123 Computer Engineering / Pikun L.V., Rosstalna O.A. Chernihiv: National University of Chernihiv Polytechnic. 32. <https://ir.stu.cn.ua/handle/123456789/30932> (2024).
- Gutiérrez, L.E., Guerrero, C.A. & López-Ospina, H.A. Ranking of problems and solutions in the teaching and learning of object-oriented programming. *Educ Inf Technol* 27, 7205-7239 <https://doi.org/10.1007/s10639-022-10929-5> (2022).
- Idris, N., Purnamawati, & Djawad, Ya. Development of Project Based Learning Model in Object Oriented Programming (OOP) Learning in Polytechnic. *Asian Journal of Education and Social Studies*, 50 (3). pp. 1-12. (2024).
- Ketsyk-Zinchenko U. Features of Professionally Oriented Foreign English Language Teaching of Students of Non-Linguistic Specialities. *Youth and Market* #9 (176) pp. 101-106 <https://pedagogy.donntu.edu.ua/2-25-2018/h-a-cherednichenko/> (2019).
- Koundilya, V., & Mathur, R. Evolution of OOP Languages *International Journal of Gender, Science and Technology Vol-10 Issue-01 June 2021* <https://ijgst.com/admin/uploads/Evolution%20of%20OOP%20Languages.pdf>
- Kubica J. Computational Fairy Tales: Classes of Cheese: Part 3 of Marcus And The Cursed Cheese <https://computationaltales.blogspot.com/search?q=CLASSES+OF+CHEESE%3A++PART+3+OF+MARCUS+AND+THE+CURSED+CHEESE+> (2011)
- Kubica J. Computational Fairy Tales: Data Validation, Marcus, and the Cheese Minstrel: Part 1 of Marcus and the Cursed Cheese. <https://computationaltales.blogspot.com/2011/08/data-validation-marcus-and-cheese.html> (2011)
- Kubica J. Computational Fairy Tales: Inheritance in Cheeses and Magic Spells: Part 4 of Marcus and the Cursed Cheese. <https://computationaltales.blogspot.com/search?q=INHERITANCE+IN+CHEESES+AND+MAGIC+SPELLS%3A++PART+4+OF+MARCUS+AND+THE+CURSED+CHEESE+> (2011)
- Kubica J. Computational Fairy Tales: Classes, Inheritance, and the Three Little Pigs. <https://computationaltales.blogspot.com/search?q=CLASSES%2C+INHERITANCE%2C+AND+THE+THREE+LITTLE+PIGS+> (2012)
- Kubica J. Computational Fairy Tales: Objects, Encapsulation, and Cheese Factories: Part 2 of Marcus and the Cursed Cheese. <https://computationaltales.blogspot.com/search?q=OBJECTS%2C+ENCAPSULATION%2C+AND+CHEESE+FACTORIES%3A++PART+2+OF+MARCUS+AND+THE+CURSED+CHEESE+> (2011)
- Object-oriented programming in Java. Lecture notes on the course 'Object-oriented programming' for students of specialties 121 – "Software engineering", 123 – "Computer engineering", 'Edited by P.G. Byvovno Chernihiv: ChNTU <https://ir.stu.cn.ua/handle/123456789/19887> (2019)
- Rangiseti, A.K. The Importance of Object-Oriented Programming. In: *Hands-On Object-Oriented Programming*. Apress, Berkeley, CA. https://doi.org/10.1007/979-8-8688-0524-0_1 (2024)
- Saidova, D. E. Analysis of the Problems of the Teaching Object-Oriented Programming to Students. *International Journal of Social Science Research and Review*, 5(6), pp. 229-234. <https://doi.org/10.47814/ijssr.v5i6.418> (2022).
- Shevchuk T. The Problem of Selecting Texts for Professionally Oriented Reading in Foreign Language Teaching and Its Role in the Formation of Students' Professionally Oriented Competence. *Modern Research in Foreign Philology*. №1 (23) <http://philol-zbirnyk.uzhnu.uz.ua/index.php/philol/article/view/307> (2023)
- Tserbuk I.F., Kosylo N.V. & Semotiuk O.V. The Specificity of Selecting Professionally Oriented Text and Creating a Complex of Lexical and Grammatical Exercises in Teaching Foreign Language to Medical Students in Higher Educational Institutions. *Actual Problems of Teaching Foreign Languages for Professional Communication* https://confcontact.com/2012edu/tom2/31_Tsebruk.htm (2012).