

ПРОГРАММНО-ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ КАК МЕТОД ФИЛОСОФСКОЙ ФОРМАЛИЗАЦИИ

SOFTWARE SIMULATION AS A METHOD OF PHILOSOPHICAL FORMALIZATION

**O. Mikhailova
D. Shlyakhta**

Summary: The article is devoted to comprehending the theoretical problematization of the meaning of constructing rationalistic models within the framework of philosophical knowledge and is aimed at substantiating the place and role of software simulation modeling in the epistemological aspect of philosophical formalization with a demonstration of the capabilities of modern programming tools in the formal philosophical analysis of the problem of the correlation of empirical and theoretical knowledge in solving paradoxes with using computer technologies to build simulation models.

Keywords: philosophical formalization, simulation in formal philosophy, rationalistic model of philosophical knowledge, Zeno's aporia, software implementation of Zeno's paradox, simulation model of experiment in C and C++.

Михайлова Ольга Николаевна

К.ф.н., доцент, Балаковский инженерно-технологический институт - филиал ФГАОУ ВО «Национальный исследовательский ядерный университет «МИФИ» (г. Балаково) molniko@mail.ru

Шляхта Дмитрий Николаевич

Балаковский инженерно-технологический институт - филиал ФГАОУ ВО «Национальный исследовательский ядерный университет «МИФИ» (г. Балаково) shlyakhta4@gmail.com

Аннотация: Статья посвящена осмыслению теоретической проблематизации значения построения рационалистических моделей в рамках философского познания и нацелена на обоснование места и роли программно-имитационного моделирования в гносеологическом аспекте философской формализации с демонстрацией возможностей современных средств программирования в формально-философском анализе проблемы соотношения эмпирического и теоретического познания в решении парадоксов с использованием компьютерных технологий построения имитационных моделей.

Ключевые слова: философская формализация, имитационное моделирование в формальной философии, рационалистическая модель философского познания, апория Зенона, программная реализация парадокса Зенона, имитационная модель эксперимента на языках СИ и С++.

Введение

Многозначные и разноаспектные проблемы континуума находят решение в философском дискурсе, который предполагает разные методы философского осмысления природы и человека с применением разных научных программ и построением соответствующих моделей как неких системно-смысловых конструкций, приближенных к наглядному оригинальному воспроизведению исследуемого предмета реальности. Современное эпистемологическое знание актуализирует философско-методологические аспекты формализации и моделирования, которые становятся в эпоху технологических вызовов постиндустриального общества третьим путём познания, дополняющим и объединяющим классическую теорию и натурный эксперимент [1]. В условиях, когда современные тенденции свидетельствуют о формировании нового поколения науки о знаниях, разрабатывающей собственные когнитивные технологии и онтологические принципы [2], формализация философского знания реализует потребность в осуществлении философских исследований прикладной проблематики с использованием метода моделирования и средств про-

граммирования [3,4,5] применительно к определенным формальным аспектам философских рассуждений.

Поскольку в рамках становления и развития первых научных программ формировались методологические принципы исследования природы и фундаментальные понятия научного мышления - понятия числа, пространства, движения, конечного и бесконечного, непрерывного и др. [6], то уже с античности для философской формализации применялись модели в качестве универсальных интеллектуальных конструкций в натурфилософских и субъективно-антропологических подходах описания/объяснения/понимания мира благодаря своей смысловой структурной композиции [7]. Существенная трансформация философско-научного мышления с его формализацией связана, как известно, с философским вкладом элейских онтологистов, осуществивших первую из известных нам в античной науке критику оснований знания, что обусловило интерес авторов статьи к рационалистическим моделям Зенона, который своими знаменитыми апориями вскрывал противоречия между чувственным и рациональным уровнями познания бытия, разрывая так называемые «зримую объективную и

умопостигаемую реальность».

Изложение основного материала статьи

Обращение к одной из наиболее известных апорий Зенона об Ахиллесе и черепахе, представляющей модель рационального доказательства, учитывает множество проведенных исследований, опровергающих или доказывающих парадокс Ахиллеса, и не претендует на обоснование «pro» и «contra», но ставит целью демонстрацию возможностей современных средств программной реализации и методов имитационного моделирования в формально-философском анализе парадокса с точки зрения решения гносеологической проблемы соотношения эмпирического и теоретического уровней познания.

В соответствии с моделью, предложенной Зеноном и исходящей из так называемого аргумента «Ахиллес», который гласит, что медленного никогда не догонит быстрый, ибо необходимо, чтобы догоняющий прежде достиг той точки, откуда стартовал убегающий, поэто-

му более медленный по необходимости всегда должен быть чуть впереди [8], посредством дихотомической формализации демонстрируется тезис, что сколько бы раз ни проходил быстроходный Ахиллес промежуток, отделявший его от тихоходной черепахи на очередном этапе, между ним и ней всегда будет оставаться разрыв в расстоянии не пройденного пути.

Анализ апории «Ахиллес и черепаха» позволяет использовать возможности процедуры программной реализации на основе, для демонстрации которой проведем детализацию содержания апории на основе следующего рассуждения: «Допустим, Ахиллес бежит в десять раз быстрее, чем черепаха, и находится позади неё на расстоянии в тысячу шагов. За то время, за которое Ахиллес пробежит это расстояние, черепаха в ту же сторону проползёт сто шагов. Когда Ахиллес пробежит сто шагов, черепаха проползёт ещё десять шагов, и так далее. Процесс будет продолжаться до бесконечности, и Ахиллес так никогда и не догонит черепаху». Таким образом, в качестве условий задачи имеются:

— скорость Ахиллеса в 10 раз превышает скорость

```

1  #include "conio.h"
2  #include "stdio.h"
3  #include "Windows.h"
4  #include "locale.h"
5
6  int main()
7  {
8      setlocale(LC_ALL, "Rus");
9      double M, N; //скорость Черепахи и Ахиллеса
10     double t; //время t
11     double xM=1000, xN=0; //начальные координаты Черепахи и Ахиллеса
12     printf_s("Введите скорость Ахиллеса = ");
13     scanf_s("%lf", &N); //Считываем скорость Ахиллеса
14     M = N / 10; //Скорость Черепахи, исходя из условия, что N = 10*M
15     printf_s("Скорость черепахи равна - %lf\n", M);
16     double R; //расстояние R
17     int i = 0;
18     while (xM >= xN) //пока координата x Черепахи больше или равна координаты Ахиллеса
19     {
20         //выполняем:
21         R = xM - xN; //высчитываем расстояние между Черепахой и Ахиллесом
22         t = R / N; //Время, за которое Ахиллес догонит Черепаху
23         xM += M * t; //Новые координаты Черепахи, с учетом изменений за время t
24         xN += N * t; //Новые координаты Ахиллеса, с учетом изменений за время t
25         printf_s("Расстояние между Ч и А - %-.20lf", R); //вывод расстояния и времени
26         printf_s("    Время, за которое А догонит Ч - %.10lf\n", t);
27
28         if ((i > 12)) system("pause");
29         i++;
30     }
31 }

```

Рис. 1. Код программы

```

Введите скорость Ахиллеса = 2
Скорость черепахи равна - 0,200000
Расстояние между Ч и А - 1000,0000000000
Расстояние между Ч и А - 100,0000000000
Расстояние между Ч и А - 10,0000000000
Расстояние между Ч и А - 1,0000000000
Расстояние между Ч и А - 0,1000000000
Расстояние между Ч и А - 0,0100000000
Расстояние между Ч и А - 0,0010000000
Расстояние между Ч и А - 0,0001000000
Расстояние между Ч и А - 0,0000100000
Расстояние между Ч и А - 0,0000010000
Расстояние между Ч и А - 0,0000001000
Расстояние между Ч и А - 0,0000000100
Для продолжения нажмите любую клавишу . . .
Время, за которое А догонит Ч - 500,0000000000
Время, за которое А догонит Ч - 50,0000000000
Время, за которое А догонит Ч - 5,0000000000
Время, за которое А догонит Ч - 0,5000000000
Время, за которое А догонит Ч - 0,0500000000
Время, за которое А догонит Ч - 0,0050000000
Время, за которое А догонит Ч - 0,0005000000
Время, за которое А догонит Ч - 0,0000500000
Время, за которое А догонит Ч - 0,0000050000
Время, за которое А догонит Ч - 0,0000005000
Время, за которое А догонит Ч - 0,0000000500

```

Рис. 2. Результат выполнения программы

черепахи;

- Ахиллес находится в 1000 шагах от черепахи;
- за время, которое Ахиллес тратит на достижения прежней позиции черепахи, она, в свою очередь, проходит расстояние, равное этому времени, умноженному на скорость черепахи (в 10 раз меньше, чем скорость Ахиллеса).

Проведем допущения: скорость Ахиллеса равна N (любое натуральное число), скорость черепахи равна $M = N/10$ – исходя из условия, что скорость Ахиллеса больше чем скорость черепахи в 10 раз; t – время, за которое Ахиллес догоняет черепаху; R – расстояние между черепахой и Ахиллесом, а xN и xM - позиции Ахиллеса и черепахи соответственно; время t , за которое Ахиллес догонит черепаху, вычисляется как расстояние между ними, разделенное на скорость Ахиллеса. Тогда, $t = (xM - xN) / N$.

Реализуем задачу, используя программу на языке СИ:

При выполнении программы у пользователя запрашивается скорость Ахиллеса, и это сделано для того, чтобы доказать, что вне зависимости от скорости Ахиллеса, он не сможет догнать черепаху.

Результат выполнения программы представлен на рис. 2.

По результатам выполнения программы видно, что с каждым выполнением действий в цикле расстояние между Ахиллесом и черепахой сокращается, но Ахиллес не может догнать черепаху, а лишь приближается к ее прежнему положению, что согласуется со смысловой конструкцией рационалистической модели Зенона с сохранением парадокса и демонстрируется рис.3.

Очевидно, что проблематично прямым эмпирическим путем опровергнуть парадокс Ахиллеса, не снимая поставленной в апории проблемы. Однако, поскольку опыт убеждает нас в том, что Ахиллес непременно дол-

жен догнать черепаху, то кажущаяся неразрешимость этого парадокса может найти своё решение через применение метода имитационного моделирования эмпирического уровня исследования проблемы как мысленного эксперимента соотношения движения самого быстрого и самого медлительного существа.

В соответствии с апорией и рассуждением на рациональном уровне, Ахиллес всегда будет отставать от черепахи на минимальное расстояние, стремящееся к 1/бесконечность, а цель Ахиллеса - перегнать черепаху [8].

Проведем имитационное решение от обратного доказательства при условии, что «Ахиллес» может перегнать «Черепаху» в тот момент, когда расстояние между ними меньше, чем шаг за одну итерацию цикла, с построением модели движения объектов – «Ахиллеса» и «Черепахи». Во-первых, у каждого из объектов есть начальная координата, в линейной системе отсчета координат - X . Условимся, что скорости этих объектов – константны.

Ахиллес – N км/ч
 Черепаха – M км/ч,
 $N > M$

Во-вторых, у каждого из объектов есть метод – «двигаться вперед», который смещает координату X на определенное значение в зависимости от его скорости в течение выполнения метода. Условие победы Ахиллеса – достижение координаты X - больше, чем у Черепахи.

Запишем алгоритм в виде цикла на псевдокоде [9]:

```

While(Ахиллес.X <= Черепаха.X) {
    Ахиллес.ДвигатьсяВперед();
    Черепаха.ДвигатьсяВперед(); }

```

Реализуем задачу, используя программу на языке C++ [10].

Разберем выполнение цикла. При каждой итерации

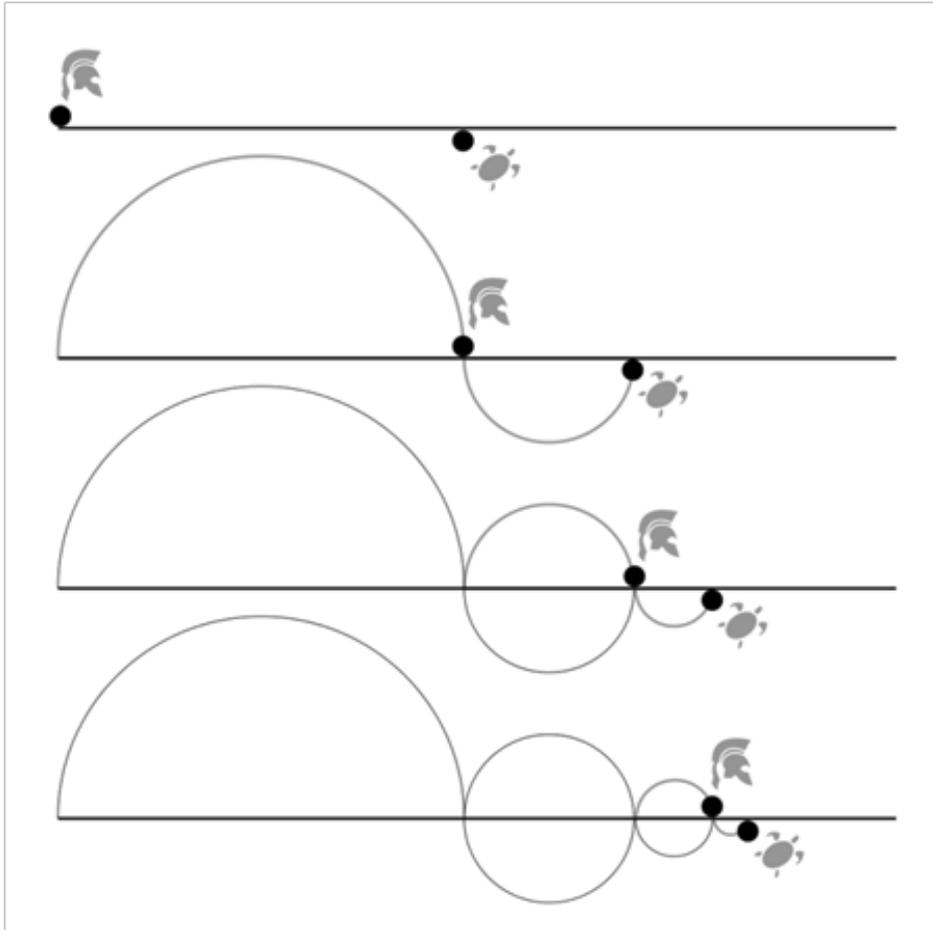


Рис. 3. Схема смысловой конструкции рационалистической модели Зенона с сохранением парадокса

```

in()

tConsoleCP(1251);
tConsoleOutputCP(1251);
tTurtle Ahilles(0, 10); //начальная координата Ахиллеса - 0, скорость -10
tTurtle Turtle(200, 1); //начальная координата Черепахи - 200, скорость -1
double curTime;
while (Ahilles.getX() <= Turtle.getX()) { //цикл выполняется до тех пор пока Ахиллес позади Черепахи
    curTime = clock(); //текущее системное время, начиная от момента старта программы
    cout << left << "Текущее время: " << setw(7) << curTime << "Координата Ахиллеса: " <<
        setw(6) << Ahilles.getX() << "Координата Черепахи: "
        << setw(6) << Turtle.getX() << endl; // вывод координат обоих объектов относительно точки
    Turtle.move(); //Черепаха совершает движение
    Ahilles.move(); //Ахиллес совершает движение

    cout << left << "Текущее время: " << setw(7) << curTime << "Координата Ахиллеса: " <<
        setw(6) << Ahilles.getX() << "Координата Черепахи: "
        << setw(6) << Turtle.getX() << endl;
    cout << "Ахиллес обогнал черепаху" << endl;
    system("pause");
}
return 0;

```

Рис. 4. Код программы

Текущее время: 494	Координата Ахиллеса: 0	Координата Черепахи: 200
Текущее время: 495	Координата Ахиллеса: 10	Координата Черепахи: 201
Текущее время: 496	Координата Ахиллеса: 20	Координата Черепахи: 202
Текущее время: 497	Координата Ахиллеса: 30	Координата Черепахи: 203
Текущее время: 498	Координата Ахиллеса: 40	Координата Черепахи: 204
Текущее время: 498	Координата Ахиллеса: 50	Координата Черепахи: 205
Текущее время: 499	Координата Ахиллеса: 60	Координата Черепахи: 206
Текущее время: 499	Координата Ахиллеса: 70	Координата Черепахи: 207
Текущее время: 499	Координата Ахиллеса: 80	Координата Черепахи: 208
Текущее время: 500	Координата Ахиллеса: 90	Координата Черепахи: 209
Текущее время: 501	Координата Ахиллеса: 100	Координата Черепахи: 210
Текущее время: 501	Координата Ахиллеса: 110	Координата Черепахи: 211
Текущее время: 502	Координата Ахиллеса: 120	Координата Черепахи: 212
Текущее время: 503	Координата Ахиллеса: 130	Координата Черепахи: 213
Текущее время: 503	Координата Ахиллеса: 140	Координата Черепахи: 214
Текущее время: 504	Координата Ахиллеса: 150	Координата Черепахи: 215
Текущее время: 504	Координата Ахиллеса: 160	Координата Черепахи: 216
Текущее время: 505	Координата Ахиллеса: 170	Координата Черепахи: 217
Текущее время: 508	Координата Ахиллеса: 180	Координата Черепахи: 218
Текущее время: 508	Координата Ахиллеса: 190	Координата Черепахи: 219
Текущее время: 509	Координата Ахиллеса: 200	Координата Черепахи: 220
Текущее время: 510	Координата Ахиллеса: 210	Координата Черепахи: 221
Текущее время: 510	Координата Ахиллеса: 220	Координата Черепахи: 222
Текущее время: 510	Координата Ахиллеса: 230	Координата Черепахи: 223

Ахиллес обогнал черепаху
Для продолжения нажмите любую клавишу . . .

Рис. 5. Результат выполнения программы

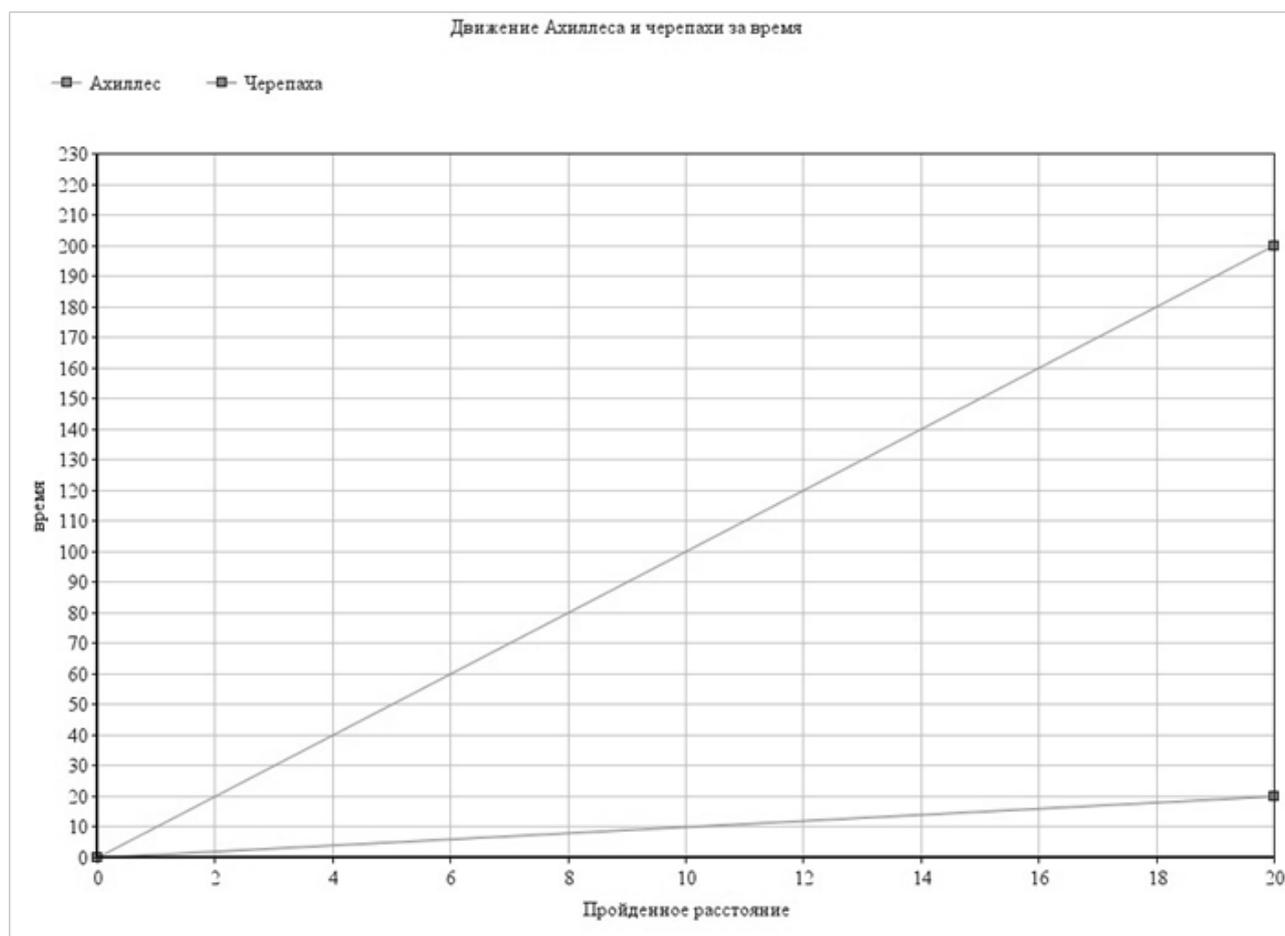


Рис. 6. Зависимость координат объектов от времени в рационалистической модели Зенона с опровержением парадокса

Ахиллес изменяет свою координату на $X = X+N$, а Черепаха $X = X+M$. После каждой итерации проверяется условие «перегнал ли Ахиллес черепаху?», если оно станет истинным, то парадокс Апории разрешен. С точки зрения математики, так как N всегда больше M , в один из моментов координаты X сравняются либо X Ахиллеса превысит X черепахи при любых начальных значениях X .

Результат выполнения программы представлен на рис. 5.

Представим зависимость координаты каждого из объектов от времени в виде графика, изображенного на рис.6.

Как видно, построение имитационной модели относительно движения быстрого и медленного объектов, воспроизводящей эмпирический уровень проверки с применением псевдокода и языка программирования C++, нарушает парадокс мысленного эксперимента и

опровергает выводы рационалистического доказательства, а апория фактически разрешается.

Выводы

Демонстрация использования программно-имитационного моделирования в контексте гносеологической проблематики формальной философии в части философского анализа парадоксов убеждает в том, что средства программной реализации и методы имитационного моделирования содержат дополнительные возможности для разностороннего и эффективного решения проблем гуманитарного знания. При этом обоснование значения философских формализмов подводит к признанию факта того, что определение приемлемости/неприемлемости программно-имитационных средств и методов моделирования в движении к истинному знанию является относительным, но весьма ценным с точки зрения акцентуализации проблем создания функциональных, устойчивых и работоспособных моделей для плодотворного философско-научного поиска.

ЛИТЕРАТУРА

1. Ильин В.П. Математическое моделирование и философия науки / В.П. Ильин // Вестник Российской академии наук. – 2018. – Т.88. – № 1. – С. 58-66.
2. Загоруйко Ю.А. Семантическая технология разработки интеллектуальных систем, ориентированная на экспертов предметной области / Ю.А. Загоруйко // Онтология проектирования. – 2015. – № 1. – С. 30-46.
3. Формальная философия. Международная лаборатория логики, лингвистики и формальной философии [Электронный ресурс]. - URL: https://lfp.hse.ru/forphil_seminar (дата обращения 07.09.2020).
4. Лобовников В.О. Философия формальная. / Гуманитарная энциклопедия: Концепты [Электронный ресурс] // Центр гуманитарных технологий, 2002–2020 (последняя редакция: 08.02.2020). - URL: <https://gtmarket.ru/concepts/7190> (дата обращения 08.09.2020).
5. Философия программирования — трёхнаправленное программирование [Электронный ресурс]. - URL: <https://habr.com/ru/post/247251/> (дата обращения 10.09.2020).
6. Гайденок П.П. Эволюция понятия науки. Становление и развитие первых научных программ. – М.: «УРСС», – 2010. – 568 с.
7. Макулин А.В. Моделирование философии: от схем, таблиц и метафор к цифровым философским визуализациям / А.В. Макулин // Исторические, философские, политические и юридические науки, культурология и искусствоведение. Вопросы теории и практики. – 2016. – № 3 (65): в 2-х ч. Ч. 1. – С. 123-127.
8. Ахиллес и черепаха [Электронный ресурс] / Электрон. дан. – М.: Wikipedia, 2019. – 1с. - URL: <https://ru.wikipedia.org/wiki/> (дата обращения 10.09.2020).
9. Структурный псевдокод [Электронный ресурс] / Электрон. дан. – М.: Tadviser, 2010. – 1с. - URL: <http://www.tadviser.ru/index.php/> (дата обращения 15.09.2020).
10. Труб И. Объектно-ориентированное моделирование на C++ [Электронный ресурс]. - URL: <https://www.livelib.ru/book/1000165129-obektnoorientirovannoe-modelirovanie-na-s-ilya-trub> (дата обращения 21.09.2020).

© Михайлова Ольга Николаевна (molniko@mail.ru), Шляхта Дмитрий Николаевич (shlyakhta4@gmail.com).

Журнал «Современная наука: актуальные проблемы теории и практики»