

5. *Jeffries A., Clandfield L., Benne R. R.* Global Upper Intermediate Coursebook. Macmillan, 2012.
6. Learner language. Learner corpora. Abstracts. — Oulu, Oulun yliopisto. October 5–6, 2012. — URL: [http://www.oulu.fi/hutk/sutvi/oppijankieli/LLLC/LLLC2012\\_abstracts.pdf](http://www.oulu.fi/hutk/sutvi/oppijankieli/LLLC/LLLC2012_abstracts.pdf)
7. *Mauranen A.* The Corpus of English as Lingua franca in Academic Settings // TESOL Quarterly. 2003. Vol. 37, No. 3. P. 515–527.

#### REFERENCES

1. *Kamshilova O. N.* Uchebnyj korpus tekstov: potentsial, sostav, struktura. SPb.: Knizhnyj Dom, 2012. 58 s.
2. *Biber D., Johansson S., Leech G., Conrad S., Finegan E.* Longman Grammar of Spoken and Written English. Harlow: Longman, 1999. P.450.
3. Cambridge Advanced Learner's Dictionary. Cambridge University Press, 2008.
4. *Hinkel E.* Simplicity without elegance: Features of sentences in L1 and L2 Academic texts // TESOL Quarterly. 2003. Vol. 37. No. 2. P. 275–300.
5. *Jeffries A., Clandfield L., Benne R. R.* Global Upper Intermediate Coursebook. Macmillan, 2012.
6. Learner language. Learner corpora. Abstracts. — Oulu, Oulun yliopisto. October 5–6, 2012. — URL: [http://www.oulu.fi/hutk/sutvi/oppijankieli/LLLC/LLLC2012\\_abstracts.pdf](http://www.oulu.fi/hutk/sutvi/oppijankieli/LLLC/LLLC2012_abstracts.pdf)
7. *Mauranen A.* The Corpus of English as Lingua franca in Academic Settings // TESOL Quarterly. 2003. Vol. 37. No. 3. P. 515–527.

*И. А. Кудрявцева*

#### АКТУАЛЬНОСТЬ РАЗВИТИЯ СОДЕРЖАНИЯ ОБУЧЕНИЯ ТЕОРЕТИЧЕСКОМУ ПРОГРАММИРОВАНИЮ

*Автором рассматривается обоснование актуальности развития содержания обучения теоретическому программированию, под которым понимается наука о математических моделях понятия «программа», и включающая следующие разделы:  $\lambda$ -исчисление, теория комбинаторов, теория типов, теория категорий, формальная семантика языков программирования.*

**Ключевые слова:** содержание обучения, теоретическое программирование, парадигма программирования, квантовая модель сознания, междисциплинарный подход, полидисциплинарный подход, трансдисциплинарный подход.

*I. Kudryavtseva*

#### The Relevance of Theoretical Programming Learning Content Development

*The development of the content of Theoretical programming course is discussed. The course deals with mathematical models of the concept of «program», and includes the following topics: lambda calculus, type theory, combinatorics theory, theory of categories, formal semantics of programming languages.*

**Keywords:** learning content, theoretical programming, programming paradigm, quantum model of consciousness, interdisciplinary approach, multidisciplinary approach, transdisciplinary approach.

Современное программирование — это достаточно развитая и сложная область научной деятельности, в которой получен ряд фундаментальных результатов, имеющих принципиальное значение для практики.

Программирование породило достаточно обширную и сложную проблематику, тесно связанную со многими разделами математики.

Для подтверждения этих слов вначале остановимся на существующем определении понятия «теоретическое программирование».

Теоретическое программирование (называемое также в ряде переводных работ теоретической вычислительной наукой, или математической теорией вычислений) [7, с. 9] — это новый раздел математической науки, чьим объектом изучения являются математические абстракции программ-предписаний, выраженных на специальных алгоритмических языках, обладающих определенной информационной и логической структурой и подлежащих выполнению на автоматических вычислительных машинах.

В. Е. Котов и В. К. Сабельфельд [10], следуя А. П. Ершову, употребляли термин «теоретическое программирование» в качестве названия математической дисциплины, изучающей синтаксические и семантические свойства программ, их структуру, преобразования, процесс их составления и исполнения.

Теоретическое программирование (при широком толковании термина) [12, с. 320] — это вся теория алгоритмов (например, теорему Геделя о неполноте можно рассматривать как теорему теоретического программирования [3]).

Теоретическое программирование (в узком смысле) [12, с. 320] — это область теории алгоритмов, концентрирующаяся вокруг взаимоотношения программы как чисто синтаксического, неинтерпретированного объекта и содержания (смысла, значения) программы.

Для теоретического программирования характерен интерес к порожденным практикой темам, которых не касалась классическая теория алгоритмов, таким как параллельное программирование или структуры данных.

Предлагаемая ниже рубрикация, раскрывающая содержание теоретического программирования, не претендует на классификационную строгость и полноту: границы между отдельными областями исследований размыты, подвижны и имеется глубокое взаимопроникновение как по изучаемым объектам, так и по методам исследований [12, с. 321].

#### 1. Математические основы программирования.

Основная цель исследований — развитие математического аппарата, ориентированного на теоретическое программирование, разработка общей теории машинных вычислений. Эти исследования тесно соприкасаются с логикой, алгеброй, теорией алгоритмов и вычислимых функций, теорией сложности вычислений, теорией автоматов, формальных языков и грамматик.

#### 2. Теория схем программ (схематология) [10].

В данной теории внимание концентрируется на изучении структурных свойств и преобразований программ, а именно тех, которые отличают программы от других способов задания алгоритмов. Главным объектом исследования становится схема программы — математическая модель программы, в которой с той или иной степенью детализации отражено строение программы, взаимодействие составляющих ее компонентов.

#### 3. Семантическая теория программ [4].

Семантика программы или отдельных конструкций языков программирования — это их смысл, математический смысл для программиста и описание функционирования для машины. Этот раздел теоретического программирования изучает методы формального описания семантики программ, семантические методы преобразований и доказательства

утверждений о программах. В частности, работы по методам проверки семантической правильности программ нацелены на автоматизацию их отладки и автоматический синтез программ.

4. *Теория вычислительных процессов и структур (теория параллельных вычислений).*

Исследования в этой области направлены на разработку и обоснование новых методов и средств программирования, прежде всего методов программирования конкурентных [17] и параллельных процессов. В частности, изучаются структуры и функционирование операционных систем, методы распараллеливания алгоритмов и программ, ведется поиск новых архитектурных принципов конструирования вычислительных машин и систем на основе результатов и рекомендаций теоретического программирования и вычислительной математики.

5. *Метрическая теория программ М. Холстеда [18].*

В основу этой теории положено приближенное комбинаторное соотношение, определяющее длину программы через число простых (различных) операторов и простых операндов. Это простое соотношение обладает удивительно мощным экстраполирующим свойством: достаточно знать, сколько будет обрабатываться данных и какими операторами, для того чтобы предсказать, какой длины будет реализующая алгоритм программа. В теории классифицируются все несовершенства программы, уменьшающие их эффективность, проводится прогнозирование времени разработки программ, числа ошибок программирования априорно, по заданному алгоритму.

6. *Прикладные задачи теоретического программирования.*

Сюда в первую очередь относятся разработка и обоснование алгоритмов трансляции [1] и алгоритмов автоматической оптимизации программ. Однако диапазон практических задач, решаемых методами теоретического программирования, постоянно расширяется.

Словосочетание «Теоретическое программирование» построено по аналогии с названиями таких наук, как теоретическая физика, теоретическая механика и т. д. В такой аналогии есть глубокий смысл: во всех случаях теоретическая научная дисциплина изучает фундаментальные понятия и законы основной науки и на основании обнаруженных закономерностей строит более частные математические модели исследуемых объектов, на которых ставит и решает прикладные задачи.

При появлении и развитии нового исполнителя программ (автоматы) акцент стал смещаться на технологии программирования в силу восприятия программы как реального объекта. Таким образом, область программирования отпочковалась от математики подобно тому, как это ранее произошло с механикой, с теоретической физикой и т. п.

Однако не следует забывать, что программирование порождает ряд понятий, методов и приемов работы, связанных именно с программами как с объектом деятельности, а не только как со средством решения исходных задач.

Сказанное подтверждает появление разнообразных парадигм программирования.

Будем придерживаться определения термина «парадигма программирования» [11] — это образец мышления по отображению класса задач в алгоритмы их решения средствами языков программирования, представляющий собой систему со следующей структурой: 1) математические основания (формальные языки), позволяющие определить особенности синтаксиса и семантики языков программирования рассматриваемой парадигмы; 2) средства, в качестве которых рассматриваются языки программирования, представляющие возможность передать алгоритм решения задачи автомату для выполнения; 3) технологии программирования, упрощающие реализацию алгоритма, с учетом архитектурной платформы, операционной системы и системы программирования.

Таким образом, необходимо более точное описание связей программирования с математикой.

Математика изучает три основных понятия: «Доказуемость», «Истинность», «Вычислимость». Для каждого понятия выделяется набор действий (операций), которые выполняет человек (Исполнитель).

В программировании выделяются те же базовые понятия, к которым добавляется понятие «Исполнитель», которым является либо автомат, либо Природа.

Теперь понятно, что появляется необходимость в анализе взаимодействий между двумя дисциплинами (математика и программирование).

В настоящее время связи между дисциплинами описываются с помощью понятия «подход», который может быть: дисциплинарным, междисциплинарным, полидисциплинарным, трансдисциплинарным.

Дисциплинарный подход позволяет выделить изолированные предметные области. Расширение области применения дисциплинарной методологии связано с накоплением дисциплинарных знаний и стремлением к полноте познания окружающего мира.

Однако развитие современной науки смещается в сторону междисциплинарного, полидисциплинарного и трансдисциплинарного подходов.

Междисциплинарность означает кооперацию различных научных областей [9, с. 249]. Примерами междисциплинарности являются биотехнологии и нанотехнологии.

Сложность объектов познания приводит к поиску ассоциаций из других областей знаний, а сходство исследуемых предметных областей обуславливает перенос методов, порождая междисциплинарные технологии и дисциплины. Тем не менее междисциплинарность предполагает сохранение дисциплинарных границ «ведущей» и «ведомой» дисциплин. Все результаты, даже те, которые получены при помощи методологии «ведомой» дисциплины, интерпретируются с позиции дисциплинарного подхода «ведущей» дисциплины.

Полидисциплинарность [9, с. 249] — это неинтегративная смесь дисциплин, в которой каждая дисциплина сохраняет свою собственную методологию и свои собственные теоретические допущения.

Наивной полидисциплинарностью мы часто пользуемся для иллюстрации математических алгоритмов.

Обнаружение новых сходств исследуемых предметных областей позволяет осуществлять полидисциплинарный подход за счет параллельного анализа объекта несколькими научными дисциплинами со своими собственными теоретическими допущениями и методологиями, не допуская дополнений извне.

Трансдисциплинарность [9, с. 250–251] характеризует такие исследования, которые идут через, сквозь границы многих дисциплин, выходят за пределы конкретных дисциплин, что следует из смысла самой приставки «транс» (тем самым создается холистическое видение предмета исследования).

Трансдисциплинарные исследования характеризуются переносом когнитивных схем из одной дисциплинарной области в другую, разработкой и осуществлением совместных проектов исследования.

В широком смысле трансдисциплинарность означает единство знания за пределами конкретных дисциплин.

Отметим, что квантовая модель сознания [13] подразумевает альтернативный выбор модели реальности, но не выбор одной из них с отбрасыванием остальных. Этот процесс предоставляет возможности предсказания эволюции реального мира.

Согласно рассматриваемой концепции, разделение альтернатив — это осознание, то есть стадия появления сознания. Снятие перегородок альтернатив происходит на «границе» сознания, когда сознание почти исчезает, отображая запутанное состояние — состояние транса.

Поэтому возникает гипотеза, что трансдисциплинарный подход — это следствие использования квантовой модели сознания.

Примерами очевидных трансдисциплинарных связей являются связи между философией и, например, математикой (так, понятие «равенство» в философии легко приводит к построению теории равенства в языке первого порядка).

Примерами неочевидных трансдисциплинарных связей являются связи рекурсивных вычислений с графикой (работы М. К. Эшер) и музыкой (И. С. Бах: каноны, фуги, ричеркары) [19].

Выделим некоторую совокупность математических дисциплин как метаязык для организации полидисциплинарной связи математики и программирования.

Гипотеза состоит в том, что этой совокупностью будет являться теоретическое программирование.

Приведем новое определение понятия «теоретическое программирование».

Опираясь на классификацию парадигм программирования [11] и их математических оснований, уделяя внимание появлению новых академических языков программирования, принимая в качестве базовых понятий «алгоритм» и «исчисление», трактуемых как способы задания конструктивных процессов, а также выделяя основополагающую для исследования математическую модель понятия «программа», предложим следующие рабочие определения понятия «теоретическое программирование».

Теоретическое программирование (гносеология) — это наука о математических моделях понятия «программа»; в частности, программа — это формула в некоторой разновидности языка первого порядка.

Теоретическое программирование (остенсивное, онтологическое) — это система, содержащая следующие разделы математики:  $\lambda$ -исчисление, теория комбинаторов, теория типов, теория категорий, формальная семантика языков программирования.

Термин «программа» является основным в курсе теоретического программирования; точнее, математическая модель понятия «программа»; в частности: 1)  $\beta$ -редукция — модель программы в функциональном программировании; 2) тройки Хоара (вместе с  $wr$ -преобразователем предикатов) — модель программы в доказательном программировании (программа — это формула в некотором расширении языка первого порядка).

Развитие содержания обучения теоретическому программированию и его реализация будут способствовать тому, что будут получены ответы на следующие основные вопросы компьютерной информатики:

- а) можно ли решить задачу с помощью автомата (или Природы);
- б) как заставить автомат (или Природу) решить задачу;
- в) как быстро автомат (или Природа) может решить задачу;
- г) можно ли решить задачу эффективно.

Теоретическое программирование играет роль «нанотехнологий» в программировании.

Использование термина связано с тем, что в настоящее время возникает конвергенция конкретных технологий в высокоинтегрированной системе, которая называется NBIC-технологиями (по первым буквам предметных областей: N — «нано», B — «био», I — «инфо», C — «когно»).

Эту же мысль Н. Н. Непейвода и И. Н. Скопин [15] выразили так: «Понятия и конструкции высокого уровня способны сократить решение и повысить его эффективность в башню экспонент раз, тогда как стандартная оптимизация или повышение мощностей техники — лишь в линейное число раз».

На наш взгляд, после изучения дисциплины «Теоретическое программирование» студенты смогут понимать содержание современных статей и монографий, посвященных вопросам теоретического программирования [2; 5; 6; 8; 14; 16].

Важно отметить, что дефицит русскоязычной литературы по основам теории языков программирования (представляющей собой важный раздел современной теоретической информатики), а также малое число переводов (приводящее к терминологическим трудностям) сказались на отсутствии в университетских курсах должного внимания к вопросам теоретического программирования.

### СПИСОК ЛИТЕРАТУРЫ

1. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции: В 2 т. Синтаксический анализ. М.: Мир, 1978. Т. 1. 614 с.
2. Берд Р. Жемчужины проектирования алгоритмов: функциональный подход. М.: ДМК Пресс, 2013. 330 с.
3. Глушков В. М. Теорема о неполноте формальных теорий с позиций программиста // Кибернетика. 1979. № 2. С. 1–5.
4. Грис Д. Наука программирования / Пер. с англ. М.: Мир, 1984. 416 с.
5. Довек Ж., Леви Ж.-Ж. Введение в теорию языков программирования. М.: ДМК Пресс, 2013. 134 с.
6. Душкин Р. Модель типизации Хиндли — Милнера и пример ее реализации на языке Haskell // Практика функционального программирования. 2010. № 5. С. 89–105.
7. Ершов А. П. Введение в теоретическое программирование. М.: Наука, 1977. 288 с.
8. Кирпичев Е. Р. Элементы функциональных языков // Практика функционального программирования. 2009. № 3. С. 83–197.
9. Князева Е. Н. Настоящее и будущее трансдисциплинарных исследований // Будущее фундаментальной науки: Концептуальные, философские и социальные аспекты проблемы. М.: КРАСАНД, 2011. С. 248–258.
10. Котов В. Е., Сабельфельд В. К. Теория схем программ. М.: Наука, 1991. 248 с.
11. Кудрявцева И. А. Методика обучения бакалавров физико-математического образования математическим основаниям парадигм программирования: Автореф. дис. ... канд. физ.-мат. наук. СПб.: Изд-во РГПУ им. А. И. Герцена, 2005. 20 с.
12. Лантев В. В., Швецкий М. В. Методическая система фундаментальной подготовки в области информатики: Теория и практика многоуровневого педагогического университетского образования. СПб.: Изд-во Санкт-Петербургского университета, 2000. 508 с.
13. Менский М. Б. Человек и квантовый мир. Фрязино: Век2, 2007. 320 с.
14. Митчелл Дж. Основания языков программирования. М.; Ижевск: НИЦ Регулярная и хаотическая динамика, 2010. 720 с.
15. Непейвода Н. Н., Скопин И. Н. Основания программирования. М.; Ижевск, 2003. 868 с.
16. Пирс Б. Типы в языках программирования. М.: Лямбда пресс; Добросвет, 2012. 656 с.
17. Хоар Ч. Взаимодействующие последовательные процессы / Пер. с англ. М.: Мир, 1989. 264 с.
18. Холстед М. Х. Начала науки о программах. М.: Финансы и статистика, 1981. 128 с.
19. Хофштадтер Д. Гедель, Эшер, Бах: эта бесконечная гирлянда. Самара: Издательский дом «Бахрах»; М., 2001. 752 с.

REFERENCES

1. *Akho A, Ulman Dzh.* Teoriya sintaksicheskogo analiza, perevoda i kompilyatsii. V 2 t. Sintaksicheskij analiz. M.: Mir, 1978. T. 1. 614 s.
2. *Bjord R.* Zhemchuzhiny proektirovaniya algoritmov: funktsionalnyj podkhod. M.: DMK Press, 2013. 330 s.
3. *Glushkov V. M.* Teorema o nepolnote formalnykh teorij s pozitsij programmista // Kibernetika. 1979. № 2. S. 1–5.
4. *Gris D.* Nauka programmirovaniya / Per. s angl. M.: Mir, 1984. 416 s.
5. *Dovek Zh., Levi Zh.-Zh.* Vvedenie v teoriyu yazykov programmirovaniya. M.: DMK Press, 2013. 134 s.
6. *Dushkin R.* Model tipizatsii Khindli-Milnera i primer ejo realizatsii na yazyke Haskell // Praktika funktsionalnogo programmirovaniya, 2010. № 5. S. 89–105.
7. *Ershov A. P.* Vvedenie v teoreticheskoe programmirovanie. M.: Nauka, 1977. 288 s.
8. *Kirpichjov E. R.* Ehlementy funktsionalnykh yazykov // Praktika funktsionalnogo programmirovaniya. 2009. № 3. S. 83–197.
9. *Knyazeva E. N.* Nastoyashchee i budushchee transdistsiplinarnykh issledovaniy / Budushchee fundamentalnoj nauki: Kontseptualnye, filosofskie i sotsialnye aspekty problemy. M.: KRASAND, 2011. S. 248–258.
10. *Kotov V. E., Sabelfeld V. K.* Teoriya skhem programm. M.: Nauka, 1991. 248 s.
11. *Kudryavtseva I. A.* Metodika obucheniya bakalavrov fiziko-matematicheskogo obrazovaniya matematicheskim osnovaniyam paradigmm programmirovaniya: Avtoref. dis. ... kand phiz.-mat. nauk. SPb.: Izd-vo RGPU im. A. I. Gertsena, 2005. 20 s.
12. *Laptev V. V., Shvetskij M. V.* Metodicheskaya sistema fundamentalnoj podgotovki v oblasti informatiki: Teoriya i praktika mnogourovnevnogo pedagogicheskogo universitetskogo obrazovaniya. SPb.: Izd-vo Sankt-Peterburgskogo universiteta, 2000. 508 s.
13. *Menskij M. B.* Chelovek i kvantovyj mir. Fryazino: Vek2, 2007. 320 s.
14. *Mitchell Dzh.* Osnovaniya yazykov programmirovaniya. M.; Izhevsk: NITs Regulyarnaya i khaticheskaya dinamika, 2010. 720 s.
15. *Nepejvoda N. N., Skopin I. N.* Osnovaniya programmirovaniya. M.; Izhevsk, 2003. 868 s.
16. *Pirs B.* Tipy v yazykakh programmirovaniya. M.: Lyambda press, Dobrosvet, 2012. 656 s.
17. *Khoar Ch.* Vzaimodejstvuyushhie posledovatelnye protsessy / Per. s angl. M.: Mir, 1989. 264 s.
18. *Kholsted M. Kh.* Nachala nauki o programmakh. M.: Finansy i statistika, 1981. 128 s.
19. *Khofshtadter D.* Gjodel, Ehsher, Bakh: ehta beskonechnaya girlyanda. Samara: Izdatelskij dom «Bakhrakh»; M., 2001. 752 s.

*А. А. Лужков, В. В. Рычгорский, И. И. Хинич*

**КОМПЛЕКСНЫЙ ПОДХОД К ФИЗИЧЕСКОМУ ПРАКТИКУМУ  
НА ВЫСОКОТЕХНОЛОГИЧНОМ ЗОНДОВОМ МИКРОСКОПЕ NANOEDUCATOR**

*Обсуждается комплексный подход к построению цикла лабораторных занятий на сканирующем зондовом микроскопе. В качестве примера рассматривается исследование дифракционных решеток и эквивалентных им квазипериодических структур. В этой работе наряду с экспериментальным изучением профиля поверхности выполняется и его математическое преобразование.*

**Ключевые слова:** физический практикум, информационные технологии образовательной среды, нанотехнологии в образовании.