**Казиахмедов Туфик Багаутдинович**,

*Федеральное государственное бюджетное образовательное учреждение высшего образования «Нижневартовский государственный университет»***\****, заведующий кафедрой информатики и методики преподавания информатики, кандидат педагогических наук, доцент,* *ktofik@yandex.ru*

**Kaziakhmedov Tufik Bagautdinovich**,

*The Federal State Budgetary Educational Institution of Higher Education «Nizhnevartovsk State University»***\****, the Head at the Chair of informatics and informatics teaching methods, Candidate of Pedagogics, Assistant professor,* *ktofik@yandex.ru*

**Симурзина Екатерина Анатольевна\***,

*преподаватель кафедры информатики и методики преподавания информатики,* *katusha-1203@yandex.ru*

**Simurzina Ekaterina Anatolyevna\***,

*the Lecturer at the Chair of informatics and informatics teaching methods, katusha-1203@yandex.ru*

**Яламов Георгий Юрьевич**,

*Федеральное государственное бюджетное образовательное учреждение высшего образования «Российская государственная академия интеллектуальной собственности», доцент кафедры информационных технологий, кандидат физико-математических наук, доктор философии в области информатизации образования, geo@portalsga.ru*

**YalamovGeorgij Yur'evich**,

*Federal State Budgetary Educational Institution of Higher Education «Russian State Academy of Intellectual Property», the Associate Professor at the Chair of information technology, Candidate of Physics and Mathematics, Doctor of Philosophy in the field of informatization of education, geo@portalsga.ru*

**МЕТОДИЧЕСКАЯ СИСТЕМА МЕЖПРЕДМЕТНОГО И МЕЖДИСЦИПЛИНАРНОГО ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ БАКАЛАВРОВ ИВТ**

**METHODOLOGICAL APPROACHES TO IMPROVING THE QUALITY OF TEACHING PROGRAMMING TO IVT BACHELORS**

***Аннотация.*** Статья посвящена анализу опыта и практики обучения программированию, содержанию и методам обучения программированию студентов ИТ направлений подготовки, которые позволяют продемонстрировать на практике возможности изучаемых языков в разработке программных комплексов и информационных систем. Предлагаемый нами подход основан на изучении не только языков, но и форм взаимодействия модулей и внешних приложений, методам реализации баз данных и моделей знаний для разрабатываемых программ. Для достижения педагогических целей авторы предлагает обучению студентов программированию на основе многоязычного подхода.

***Ключевые слова:*** стили программирования; многоязычное программирование; взаимодействие модулей приложений; активизация внешних модулей; сетевое программирование.

***Annotation****.* The issue of developing programming skills in high-level languages remains relevant in our time. High-level programming has given rise to many programming languages and styles. But the information systems themselves are focused on data management on the Web server, on the database server and on the file server. At the same time, the class of tasks to be solved requires knowledge of not one high-level language, but a whole set of them. It is very important to note that the studied classical languages Pascal, C++ have a rather meager set of composite data, unlike Java, JavaScript, Python, etc. On the other hand, multilingualism allows you to form skills of working with data created by means of one language using data management methods of other languages.

It is necessary to take into account the fact that all high-level languages allow you to write programs in logical, functional, object-oriented styles. The issues of forming programming styles can be implemented in different languages, but the organization of interaction between data, their processing and storage in the necessary formats requires the use of other languages. Therefore, it is important that students at the beginning of the path have an understanding of both styles, programming methods, and data formats for subsequent processing using modules of the system being developed or using appropriate basic data management libraries in programming environments and languages.

***Keywords****:* programming styles; multilingual programming; interaction of application modules; activation of external modules, network programming.

Целью изучение высокоуровневого программирования бакалаврами по направлению «Информатика и вычислительная техника» является формирование мышления программиста через представление алгоритмов в виде процедур, функций, предикатов как свободных, так и являющихся методами классов, а также методами управления данными через разрабатываемые алгоритмы.

При реализации задач достижения указанной цели, необходимо обратить внимание на такие компоненты алгоритмического мышления как рекурсивные функции и процедуры, логические функции(предикаты), методы, привязанные к классам, копии которых мы размещаем в памяти. Причем здесь важно ориентировать студентов, что алгоритмы позволяют построить программы логическим, функциональным, объектно-ориентированным стилями программирования.

Существующие методики обучения программированию в основном ориентированы на изучения алгоритмических конструкций, базовых типов и структур данных. В реальной жизни структуры данных весьма разнообразнее и базовые типы, и структуры нам позволяют создавать необходимые для конкретной задачи сложные структуры данных. Механизмам их создания, хранения и управления также необходимо уделить особое внимание.

В наше время мы являемся свидетелями исчезновения существующих и появления новых профессий. Это накладывают обязательные изменения и в стандарты обучения по IT направлениям. Мы даже видим, что со временем изменится функционал бухгалтерии, экономических отделов предприятий, основные функции будут выполнять программные комплексы. А это с нашей точки зрения, требует подготовки будущих IT специалистов в правильности формирования актуальной информационной базы и составления автоматизированных отчетов предприятий.

Знание одного какого-либо языка высокого уровня при изучении дисциплины «Программирование» является с нашей точки зрения не эффективным по следующим причинам:

* Не всегда инструментарий языка достаточен для манипуляции сложными типами и структурами данных.
* Не содержит библиотек по созданию эффективных баз данных, их размещению в памяти, в том числе, и в Heap-области и манипуляции внешними базами данных.
* Не позволяют программировать использование данных в различных форматах их кодирования и хранения. Например, данные хранимые в HTMLфайлах и отображаемые на Web страницах, данные на внешних серверах и прочее.
* Знание одного языка не позволяет выработать правила выбора стиля программирования применительно к классу решаемых задач: вычислительные, логического вывода (алгоритмы экспертизы), алгоритмы поиска, алгоритмы взаимодействия в сети и с приложениями и т.п.

Знание множества языков не является признаком качественного усвоения дисциплины, так как о качестве знаний в области современного высокоуровневого программирования можно говорить, если обучение строится на принципе межпредметности знаний в современном обществе (междисциплинарность образования).

Взаимопроникновение дисциплин особенно важно учитывать при обучении программированию. Все современные науки в той или иной степени имеют информационные инструменты организации научных экспериментов и вычислений. Даже это требует наличия алгоритмической культуры у специалистов, занимающихся наукой в не ИТ сфере. Но ввиду того, что для научных отделов, для коллективов ИТ фирм и предназначено подготовка бакалавров по данному направлению и государство уделяет этому особое внимание, необходимо при изучении данного курса ориентировать в какой-то степени студентов на такие области как химия, биология, мехатроника и робототехники, математика, физика, экономика, бухгалтерия и др.

Для эффективного обучения программированию необходимо учитывать то, что для изучения самих языков программирования необходимо знание алгебра логики, теории множеств и реляционной алгебры, матричной алгебра, основ схемотехники и физики. Очень важно обратить внимание на учет особенностей файловых(дисковых) операционных систем, форматов хранения данных для организации в программах работы с данными.

Такое содержание данного курса мы связываем с тем, что развитие информатики и информационных технологий происходит на стыке множества наук, в том числе перечисленных выше разделов. В решении проблемы изучения большого объема знаний можно использовать компоненты дополнительного образования, которое сегодня прокладывает себе дорогу в высших учебных заведениях. Поэтому лабораторные работы должны быть межпредметными, надпредметными, а это потребует самостоятельного изучения студентами инвариантных компонент других научных сфер.

Использование робототехники, роботов как исполнителей алгоритмов является одним из важных методик обучения программированию. Поэтому знания в области робототехники и мехатроники должны и входить в состав профессиональных компетенций бакалавра информатики и это касается бакалавров – будущих учителей информатики. В системе подготовки бакалавров данного направления в нашем вузе обязательным является изучение дисциплины «Образовательная робототехника» [1; 2; 3; 9].

Алгоритмические конструкции всех языков высокого уровня идентичны, но на этой основе нельзя построить предположение, что для данного курса достаточен один язык высокого уровня. Все серьезные отличия языков связано организацией сложных типов данных, как списки, массивы, множества, таблицы, методы организации связей между данными и т.п. Поэтому для реализации профессиональных проектов знание одного языка недостаточно, так как сами проекты, методы представления данных нельзя реализовать качественно средствами одного языка. Конечно, при изучении программированию нельзя оставить без внимания формирование интерфейсов приложений.

Многоязычие дает бакалавру знания о том, что разные языки позволяют включить в среду дополнительные свободно распространяемые библиотеки разного уровня.

Изучая алгоритмы, необходимо обучение их представления на языках высокого уровня при реализации конкретных задач программирования.

Формирование алгоритмического мышления основано на применении разных стилей программирования для выполнения конкретных лабораторных длительного характера, профессиональных проектов.

При анализе инструментария реализации программ студенты понимают о необходимости самостоятельного изучения некоторых вопросов вне содержания дисциплины и с этими вопросами выступают на семинарских занятиях, посвященных особенностям языков программирования. На этом курсе ведется и подготовка к изучению визуального программирования, событийное программирование изучается, рассматривая разработку программ на JavaScript.

Приведем содержание программы и методики ее достижения, хотя об организации междисциплинарного многоязычного подхода в обучении бакалавров ИВТ более подробно можно ознакомиться в [4; 6].

И так, наш курс является 2 годичным, объем 18- ЗЕ (648 часов-1, 2, 3, 4 семестры) и содержит следующие 9 разделов:

1. Среды и языки программирования. Типы данных. Реализация линейных алгоритмов.
2. Ветвление и выбор. Продукционные модели знаний.
3. Циклические алгоритмы в языках высоко уровня PascalABC, Java, C++, JavaScript, PHP, Питон
4. Модульное программирование в ЯВУ PascalABC, Java, C++.
5. Процедурное и функциональное программирование в языках высоко уровня PascalABC, Java, C++, Питон, JavaScript, PHP. Процедурные знания.
6. Методы организации данных. в языках высоко уровня PascalABC, Java, C++, Питон, JavaScript, PHP
7. Объектно-ориентированное программирование в языках высокого уровня. PascalABC, Java, C++, Питон, JavaScript, PHP.
8. Графика в языках высоко уровня PascalABC, Java, C++, JavaScript, PHP.
9. Разработка программ с использованием нескольких языков высокого уровня.

При построении содержание, как видно, из наименования разделов, мы связываем с каждым разделом методы представления данных, знаний. Хочется акцентировать внимание на том, что, изучая программирования не только изучаются и закрепляются алгебра логики, матричная алгебра, реляционная алгебра, методы хранения данных, методы обработки данных. Для качественного понимания нашего курса студентам приходится углубляться в теорию баз данных, сетей, сетевого программирования, что является очень актуальным для подготовки ИТ кадров для цифровой экономики.

Если обратить внимание на последовательность изучения разделов, то хочется отметить следующее: каждый тип алгоритмической конструкции связан с какой- либо моделью знаний.

Например, продукционные модели знаний рассматриваются в разделе изучения ветвления и выбора, в разделе 5 рассматривается процедурные знания. При этом отмечаем, что многие базовые библиотеки в классических языках высокого уровня реализованы в виде процедур и функций, а в объектно-ориентированных языках библиотеки представлены в виде классов и процедурные знания являются частью или методами классов.

В курсе рассматривается модульная организация приложений, которая включает в себя и привлечение чужих внешних модулей. Например, при разработке обучающих программ сами языки выступают в роли внешних модулей. Набрав программу в конкретном текстовом окне, отправляем ее на исполнение т.е. передаем текст программы на исполнение.

Потребность в использовании разных стилей программирования связано с набором задач для программирования. С первых дней все же необходимо научить студентов к выбору оптимального алгоритма решения задачи.

Рассмотрим простейшие задачи, решение которых позволит использовать функциональное (в том числе и рекурсивное), логическое программирование.

**Задание на использование функций:**

1. Создайте функцию, заменяющую в исходном массиве первое вхождение заданного значения другим.
2. Создайте функцию, заменяющую в исходном массиве все вхождения заданного значения другим.
3. Создайте функцию, порождающую по заданному натуральному числу N массив (динамический), состоящий из натуральных чисел от 1 до N (по возрастанию).

Приведем Коды и рабочие экраны (рис.1, рис. 2).

**

*Рис. 1. Листинг программы*

Для создания предикатов можно также использовать JavaScript, но можно и использовать Prolog.

Второй тип задач, это задачи на организацию взаимодействия приложений, реализованных в разных средах, а также хранение данных и их чтение из файлов различных стилей. Например, данные поступают в приложение в HTML формате, а нужно хранить в виде базы данных. Именно такой подход дает полное понимание современного программирования и технологий разработки приложений [5].

Третий тип задач охватывает преобразование форматов данных, создание собственных форматов, вывод графики, ускорение работы с графикой.

Четвертый тип задач, это разработка игр, способствующих пониманию современных технологий программирования в игровой сфере, многоагентного программирования [7; 8].

Пятый тип задач охватывает серверное программирование.

По сложности практических и лабораторных работ можно привести следующую классификацию:

* погружающие задачи в среды и языки программирования;
* задачи программирования робототехнических систем (совместно с преподавателем робототехники);

*Рис. 2. Окно браузера с выполненными скриптами*

* задачи c использованием разных стилей программирования;
* программирование модулей для размещения. хранения и манипуляции большими объемами данных;
* клиент-серверное программирование манипуляции данными;
* клиент-серверное web программирование;
* программирование мобильных устройств;
* задачи на знание основ сетевого программирования;
* задачи с длительным сроком реализации для решения задач автоматизации для реальных предприятий.

Конечно, все эти задачи носят в основном ознакомительный, погружающий дальнейшее углубление происходит за счет изучения специальных дисциплин и дисциплин по выбору.

*Литература*

1. Ваграменко Я.А., Казиахмедов Т.Б., Яламов Г.Ю. Методическое обеспечение подготовки учителей образовательной робототехники. Педагогико-технологический аспект // Педагогическая информатика. 2016. № 1. C. 30-44.
2. Ваграменко Я.А., Казиахмедов Т.Б., Яламов Г.Ю. Методическое обеспечение подготовки учителей образовательной робототехники. Методический аспект // Педагогическая информатика. 2016. № 2. C. 41-50.
3. Образовательная робототехника как инновационная технология обучения: монография // Я.А. Ваграменко, О.М. Карпенко, Г.Ю. Яламов [и др.] / Открытый гуманитарно-экономический университет, БИОР УМНЕЙ, Институт управления образованием РАО. Москва: Издательство Современного гуманитарного университета, 2019. 105 с.
4. Казиахмедов Т.Б., Симурзина Е.А. Междисциплинарный подход при изучении типов и структур данных в языках // Современное программирование: Материалы IV Международной научно-практической конференции [г. Нижневартовск, 8 декабря 2021] / Под общей редакцией Т.Б. Казиахмедова. Нижневартовск: Нижневартовский государственный университет, 2022. С. 384-392. DOI 10.36906/AP-2022/64.
5. Казиахмедов Т.Б., Мосягина Т.В. Формирование web компетенций бакалавров ИВТ в условиях наличия большого набора инструментов для web проектирования // Современное программирование: Материалы IV Международной научно-практической конференции [Нижневартовск, 8 декабря 2021] / Под общей редакцией Т.Б. Казиахмедова. Нижневартовск: Нижневартовский государственный университет, 2022 С. 439-445. DOI 10.36906/AP-2022/72.
6. Казиахмедов Т.Б., Пащенко И.О., Яламов Г.Ю. Методические подходы обучения парадигмам и методам программирования будущих бакалавров по IT-направлениям // Педагогическая информатика. 2022. № 3. С. 217-228.
7. Казиахмедов Т.Б., Симурзина Е.А. Особенности обучения программированию бакалавров, обучающихся по направлению «информатика и вычислительная техника» // Педагогическая информатика. 2023. № 1. С. 222-229.
8. Горлов С.И., Казиахмедов Т.Б. Методы обучения программированию IT бакалавров: игровые среды и программирование игр // Педагогическая информатика. 2023. № 2. С. 191-195.
9. Применение программно-управляемых устройств в профильном обучении (методические рекомендации) // Я.А. Ваграменко, О.А. Шестопалова, Т.Б. Казиахмедов Г.Ю. Яламов / Педагогическая информатика. 2014. № 1. С. 3-21.

*Источник: Казиахмедов Т.Б., Симурзина Е.А., Яламов Г.Ю. // Педагогическая информатика. 2023. № 4.*