

ТЕХНИЧЕСКИЕ НАУКИ

Касумова А.М.

Алыева Э.З

*Азербайджанский Государственный Педагогический
Университет, г. Баку*

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЯЗЫКОВ PYTHON, C++ И JAVA ДЛЯ ОБУЧЕНИЯ СТУДЕНТОВ ПРОГРАММИРОВАНИЮ

Kasumova A. M.

PhD in pedagogy,

Alyeva E.Z.

Azerbaijan State Pedagogical University, Baku

COMPARATIVE ANALYSIS OF PYTHON, C ++ AND JAVA LANGUAGES FOR TEACHING STUDENTS IN PROGRAMMING

Аннотация. Статья посвящена изучению проблемы выбора языка программирования при обучении студентов основам программирования. Рассматриваются различные методики обучения программированию и языки программирования, используемые в рамках данных методик. Анализируется популярность современных языков программирования и возможность использовать их в качестве основного языка при обучении программированию студентов в рамках базового курса информатики.

Abstract. The article is devoted to the study of the problem of choosing a programming language when teaching students the basics of programming. Various methods of teaching programming and programming languages used in the framework of these methods are considered. The article analyzes the popularity of modern programming languages and the possibility of using them as the main language in teaching programming to students within the framework of a basic computer science course.

Ключевые слова: программирование, языки программирования, обучение программированию, базовые курсы.

Key words: programming, programming languages, teaching programming, basic courses.

Само существование сетевых технологий изменило природу образовательного процесса. Современные сетевые технологии улучшают способность общения каждого человека и предоставляют людям во всем мире небывалый доступ к информации. Сегодня они стали необходимым фундаментом компьютерной науки, поэтому невозможно представить себе программу обучения информатике, в которой этой теме не уделялось бы значительное внимание. В большинстве учебных программ на сегодняшний день – не только в информатике, но также и в других областях – сетевые технологии стали важным педагогическим инструментом. Поскольку при проектировании нового поколения образовательных стандартов естественно исходить из самых последних достижений информатики и существующих тенденций ее развития, то необходимо кратко проанализировать, какие аспекты информатики изменились за время, прошедшее после принятия нынешнего поколения стандартов. Все эти изменения делятся на две категории, техническую и культурную, каждая из которых оказывает существенное влияние на образование в сфере информатики.

Образовательный курс информатики состоит из нескольких содержательных линий: информация; формализация и моделирование; алгоритмизация и программирование; компьютер;

информационные технологии. Основной предметной целью раздела «Алгоритмизация и программирование» является формирование у студентов умения использовать языки программирования для моделирования ситуаций и решения задач. Решение алгоритмических задач различного уровня сложности позволяет оценить не только уровень предметных навыков и умений, но и уровень развитости познавательной самостоятельности студентов.

Компьютеры чаще всего используются для различных целей в инженерии и науке, включая управление, анализ данных, моделирование и оптимизацию дизайна.

Программирование - это фундаментальная способность для специалистов по информатике. В мире существует более тысячи разных языков программирования. Некоторые из них чуть меньше, чем полностью бесполезны. У языков программирования есть одна особенность — если два языка имеют сходство в парадигмах, то, как правило, они схожи и в синтаксисе.

Парадигма является инструментом грамматического описания фактов, явлений и процессов, событий, возможно, не существующих одновременно, но интуитивно объединяемых в общее понятие. Парадигма представляет и определяет то, как программист видит выполнение плана решения поставленной программы.

Нужно отметить, что парадигма программирования не определяется однозначно языком программирования; практически все современные языки программирования в той или иной степени допускают использование мультипарадигмальное программирование. Точки зрения создателей языков гласит, что ни одна парадигма не может быть одинаково эффективной для всех задач, и следует позволять программисту выбирать лучший стиль программирования для решения каждой отдельной задачи.

В учебной программе по программированию используются традиционные языки, такие как C, C++ или Java, в качестве первого языка обучения для студентов, которые эффективны для разработки реальных приложений и поэтому популярны в промышленности. Тем не менее сложный синтаксис этих языков является проблемой для новичков, что становится препятствием для обучения студентов.

Несмотря на единые требования образовательного стандарта, у каждого студента свой набор требований к языку программирования. Это может быть удобная, дружелюбная среда разработки; простой, интуитивный синтаксис; кроссплатформенность и другие. Все это не относится к методике, но является важным фактором, который влияет на выбор языка. Чтобы обосновать требования к языку программирования, необходимо, в первую очередь, определить цель изучения программирования. К сожалению, наличие четких целей не дает нам единого ответа на вопрос, какой язык программирования лучше изучать студентам, так как многие из современных языков программирования легкостью удовлетворяют каждый из указанных целевых аспектов.

Python имеет более простой синтаксис и структуры данных высокого уровня, позволяющие создавать сжатые программы. Множественные парадигмы Python также предоставляют студентам возможность изучать различные особенности языков программирования. Таким образом, Python постепенно становится новым вариантом первого языка для изучения. Учитывая вышесказанное можно обратить внимание на особенности первого обучаемого языка программирования путем сравнения Python с популярными языками обучения C++ и Java.

Язык программирования C++ отличается эффективностью как объектно-ориентированный язык; хотя он поддерживает некоторые элементы функционального программирования. Доказано, что «разрыв в скорости между Java и C++ можно объяснить более медленными методами управления памятью в виртуальных машинах Java». Это началось как расширенное улучшение языка C. Он использует файлы заголовков и директивы препроцессора. Его выполнение программы обычно начинается с функции *main ()*. Это очень быстрый язык программирования, так как большинство его модулей написаны на

машинном языке и могут эффективно работать на определенных машинах. Он включает в себя набор функций как высокого, так и низкого уровня, упрощающих аппаратную реализацию. Это зависит от системы; однако для реализации могут использоваться определенные библиотеки и системные утилиты. С точки зрения безопасности этот язык уязвим для взлома указателей. Он менее близок к естественному языку по сравнению с Java. Хотя C++ занимает сравнительно меньше количество строк кода. C++ не хватает эффективности управления памятью по сравнению с некоторыми другими языками, такими как Java.

Старые версии Java, имеют более длительное время выполнения по сравнению с таким языком, как C++; хотя реализации неуклонно улучшаются с использованием компиляции JIT (Just-In-Time) и таких областей, как процедуры сборки мусора. Это язык, в котором есть интерпретаторы и компиляторы, причем первые относительно медленнее. Java и C++ имеют много семантического и семантического сходства. Доказано, что время выполнения Java примерно в 2 раза медленнее, чем время выполнения C++.

Java унаследован синтаксис от C/C++, хотя первый почти полностью объектно-ориентирован. Большинство его примитивных типов являются функциями, значения в основном являются объектами, а переменные принадлежат на занятия. Программы не зависят от платформы. Программы сначала компилируются JVM в промежуточный язык, называемый байт-кодами, а затем в коды машинного языка. Его можно охарактеризовать как переносимый язык так как, написанное на Java приложение можно запустить на любой платформе, если на ней установлена среда исполнения Java (*JRE, Java Runtime Environment*).

Этот язык использует автоматический сборщик мусора или *garbage collector* в качестве фонового потока, чтобы гарантировать, что память доступна по запросу для достижения высокой производительности. Когда созданные объекты разработчикам перестают использоваться, JRE с помощью сборщика мусора очищает память. Также, имеет надежную систему безопасности. Программисты могут использовать набор API для достижения высокого уровня безопасности. Основную сущность языка Java составляют библиотеки файлов, называемые классами, каждый из которых содержит небольшие фрагменты проверенного, готового к выполнению кода. JVM как менеджер безопасности также удаляет не доверенные коды из операционной системы. Он использует потоки и некоторые языковые расширения для продвижения параллельного программирования. Сокеты - это низкоуровневые интерфейсы, используемые в сетевом программировании для потоковой передачи данных.

Java относится к языкам программирования общего назначения. Студентам нужно знать, все

возможности Java, чтобы определиться с выбором языка.

Язык Java считается дружелюбным для начинающих. Но, он не такой лаконичный, как Python. Однако создатели Java стремились сделать язык простым и лёгким для изучения.

Python наиболее приближен к алгоритмическому языку программирования. Не имеет строгих синтаксических правил, является более современным языком программирования, чем вызывает больший интерес у студентов. Язык Python считается языком высокого уровня и для решения математических задач. На первый взгляд очень похожий на MATLAB: интерпретируется, имеет интерактивную подсказку, позволяет динамическую типизацию и обеспечивает автоматическое управление памятью (и имеет встроенный комплекс чисел).

Python дает несколько преимуществ над MATLAB в контексте обучения: Python имеет очень четкий, недвусмысленный и интуитивно понятный синтаксис и использует отступы для группировки блоков операторов; имеет небольшое ядро команд, которые обеспечивают почти все функции, которые потребуются студентам; Python можно использовать как полностью объектно-ориентированный язык, который поддерживает разные стили кодирования; интерпретатор Python - это бесплатное программное обеспечение (т.е. легкодоступное), которое существует практически для всех платформ (включая Windows, Linux / Unix, Mac OS).

Стоит отметить, что хотя Python довольно молодой язык, это относительно стабильный язык, который все чаще используется в промышленности и академических кругах (Philips, Google, NASA, ВМС США и Disney). Он также обеспечивает основу для создания больших модульных кодов и управления ими.

Из анализа поисковых запросов можно выяснить, что у Java и C++ были взлёты и падения. А рейтинг Python постепенно рос, он стал чуть ли не самым популярным языком программирования.

В вузах обычно учат языкам программирования в рамках предметов по

информационным технологиям, которые часто выглядят как дополнения к математическим курсам, либо привязаны к некоей основной специальности, скажем, к электротехнике.

Как не странно, большинство учебных заведений ставят знак равенства между программированием и информационными технологиями, а информационные технологии приравнивают к математике.

Многие программисты согласятся с тем, что высокоуровневые скриптовые языки сравнительно легки в изучении. Как обычно они утверждают, что C++ отлично подходит для разработки игр, а Python нет равных в научных и статистических расчётах и конечно же Java пригодится тем, кто видит себя программистом в крупной компании.

В конце можем сделать заключение, что ключ к успеху - практика, создавайте свои проекты.

Литература:

1. Kernighan, B.W., Ritchie, D.M.: The C Programming Language. Prentice Hall Software Series (1988)
2. The Mathworks: Matlab (2003) www.mathworks.com.
3. Prechelt, L.: An empirical comparison of seven programming languages. IEEE Computer 33 (2000) 23–29
4. van Rossum, G.: Python tutorial. Centrum voor Wiskunde en Informatica (CWI), Amsterdam. (1995) www.python.org.
4. Абельсон Х., Сассман Д. Структура и интерпретация компьютерных программ. — М.: Добросвет, 2010
5. Donaldson, T.: Python as a first programming language for everyone. In: Western Canadian Conference on Computing Education. (2003) www.cs.ubc.ca/wccce/Program03/papers/Toby.html.
6. Beecher Karl. Computational Thinking: A Beginner's Guide to Problem-Solving and Programming, BCS, 2017
7. Ben-Ari M. Understanding Programming Languages, John Wiley & Sons, 1996.
8. Страуструп Б. Язык программирования C++. 3-е изд. — М.: Бином, 2011.