

УДК 004.43

ТЕНДЕНЦИИ РАЗВИТИЯ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

Салова Т.Л.,

*кандидат технических наук, доцент,
кафедра Информационных технологий и математики,
Сочинский Государственный Университет,
г. Сочи, Россия*

Хеладзе Л.Д.,

*магистрант,
Сочинский Государственный Университет,
г. Сочи, Россия*

Аннотация

В исследовании представлена информация о программной инженерии (SE) и сферах знаний, охватываемых ею, исследуются этапы разработки программирования и проблемы, возникающие в этой области. Проанализирована информация о языках программирования, проведен их сравнительный анализ и разъяснены преимущества и недостатки каждого из них. Представлены статистические данные по рейтингу языков программирования.

Ключевые слова: программная инженерия; программирование; языки программирования; рейтинговый список; алгоритм.

TRENDS IN THE DEVELOPMENT OF PROGRAMMING LANGUAGES

Salova T.L.

*candidate of Technical Sciences, an associate professor,
Department of Information Technologies and Mathematics,
Sochi State University,
Sochi, Russia*

Kheladze L.D.

undergraduate

Sochi State University,

Sochi, Russia

Annotation

The study provides information about software engineering (SE) and the areas of knowledge covered by it, examines the stages of programming development and the problems arising in this area. The information about programming languages is analyzed, their comparative analysis is carried out and the advantages and disadvantages of each of them are explained. Statistical data on the rating of programming languages are presented.

Keywords: software engineering; programming; programming languages; rating list; algorithm.

Программная инженерия понимается как наука, которая систематизировала и упорядочила методы для разработки, тестирования, обслуживания и использования программного обеспечения.

Разработка высококачественного программного обеспечения - это процесс, требующий больших усилий. Процессы, инструменты, технологии и человеческие ресурсы должны взаимодействовать друг с другом для высококачественной обработки необходимого проекта. Одной из главных целей современной разработки программного обеспечения является создание программных продуктов для различных проблемно-ориентированных областей и обеспечение их эффективности на протяжении всего жизненного цикла.

Разработка программного обеспечения включает в себя следующие области знаний:

- основы вычислительной техники;

- основы математики и инженерного дела;
- профессиональный опыт (командная работа, коммуникативные навыки);
- основы моделирования (анализ, работа с требованиями, спецификациями);
- проектирование программного обеспечения (концепции и стратегии проекта, проектирование человеко-машинного интерфейса, поддержка проекта);
- проверка и аттестация программного обеспечения (тестирование, оценка пользовательского интерфейса);
- основы эволюции программного обеспечения;
- процессы разработки программного обеспечения;
- качество программного обеспечения;
- управление проектами (концепции управления, планирование и отслеживание проекта, внедрение, управление персоналом и конфигурация).

Стандарты в этих областях точны и четко описаны. При создании программного обеспечения для решения этих задач используются различные языки программирования [2]. Главным требованием в программировании является создание алгоритмов. Алгоритм - это последовательность операций (этапов), которые необходимы для решения проблемы. Как правило, алгоритм представляет собой формальную запись, которая определяет операции, необходимые для решения данной проблемы, и последовательность, в которой они выполняются. Процесс создания алгоритмов называется алгоритмизацией.

Тенденции развития языков программирования следуют некоторым фундаментальным принципам. Это способность к будущему совершенствованию, сосредоточенность на эффективности программирования, способность учитывать сложность проблемы и продлевать жизненный цикл программы. Рассмотрим следующие этапы развития языков программирования [3]:

Первые языки программирования

Эти языки были созданы в 1940-х – 1950-х годах, в которых программа состояла из линейной последовательности элементарных инструкций «низкого уровня». Преимущества первых языков программирования заключаются в том, что они обладают высокой вычислительной эффективностью. Однако они существенно зависят от вычислительной среды. Некоторые из примеров таких языков программирования включают машинные коды и язык ассемблера.

Процедурные (императивные) языки программирования

Эти языки были созданы в 1950-1970-х годах. Эти языки могут включать подпрограммы, которые вызываются в любое время во время выполнения произвольной процедурной программы. Программы используют глобальные и локальные переменные. Некоторые из преимуществ этого класса языков программирования включают более высокие уровни абстракции, низкую зависимость и широкую совместимость. Тем не менее, основными проблемами являются большие трудозатраты, низкая эффективность кода. Примерами таких языков программирования являются FORTRAN, ALGOL, PL/1, APL, БПЛ, COBOL, Pascal C и Basic.

Декларативные языки программирования

Эти языки были созданы в 1960 году. Как правило, программы описываются в простых терминах. Простота верификации программного обеспечения и тестирование, легкость форматирования, высокая абстракция были основными преимуществами этих языков программирования. С другой стороны, трудность осознания потребности в фундаментальных математических знаниях препятствует развитию языков. Примерами таких языков программирования являются LISP (Interlisp, Common Lisp, Scheme), SML, Haskell и Prolog.

Функциональные языки программирования

Примерами функциональных языков программирования являются SML, CaML, Haskell, Miranda и Hope, которые были созданы в 1960-х годах. Дневник науки | www.dnevniknauki.ru | СМН ЭЛ № ФС 77-68405 ISSN 2541-8327

Преимущества функциональных языков программирования заключались в том, что они полностью автоматизировали управление памятью компьютера, простоту повторного использования фрагментов кода и охватывали широкий спектр параметрических функций (параметрический полиморфизм). Недостатки связаны с их нелинейной структурой и относительно низкой эффективностью.

Логические языки программирования

Фундаментальной особенностью является то, что присутствует набор правил, состоящий из логических команд. Созданные в 1970 году логические языки программирования обладают высокоуровневой абстракцией, удобной логической ориентацией программирования и широко применяются в экспертных системах.

Ограниченная доступность и нелинейная структура программного обеспечения являются основными недостатками, которые необходимо преодолеть. Примерами таких языков программирования являются Prolog и Mercury.

Объектно-ориентированные языки программирования

Как следует из названия, программа описывает объекты, агрегаты, отношения и методы взаимодействия между объектами. Преимущества этого типа языков программирования заключаются в простоте использования при моделировании, близости к предметной области, ориентации на идентифицируемое событие, в высоком уровне абстракции и легком повторном использовании фрагментов программного кода. Основной проблемой этого типа языков программирования является сложность проверки и тестирования программ. Примерами таких языков программирования являются C++, Visual Basic, C#, Eiffel и Oberon, которые были разработаны в 1970-1980 годах.

Скриптовые языки программирования

Эти языки были созданы в 1990 году и обладают способностью повторно использовать возможные сценарии и код. Благодаря этой особенности они обладают преимуществами близости к предметной области, высокой степенью

абстракции и высокой мобильностью. Однако трудно проверять и тестировать программы, что характерно для всех объектно-ориентированных языков программирования. К этому классу относятся: VBScript, PowerScript, LotusScript и JavaScript.


Языки параллельного программирования

Примерами языков параллельного программирования, созданных в 1980 году, были Ada, Modula2 и Oz. Из-за высокой вычислительной эффективности для больших систем эти языки применялись в одно и то же время. Слабые стороны языков заключаются в их относительно низкой скорости работы и больших затратах на эксплуатацию.


В настоящее время существует множество языков программирования, и они используются при разработке программных продуктов. Возникает такой вопрос: нужны ли новые языки программирования?


Дело в том, что, хотя языки программирования являются мощными и многофункциональными, синтаксис ни для одного из этих языков в настоящее время не является совершенным и универсальным. Помимо этого, языки программирования постоянно развиваются. Широковещательные многоядерные процессоры, облачное программирование, портативная техника (цифровая видеокамера, музыкальный проигрыватель, мобильные телефоны, GPS-навигаторы, ноутбук, iPad и т.д.), а также распределенная архитектура создали новые проблемы для производителей. Добавление новых функциональных возможностей, парадигм и шаблонов к существующим языкам программирования, которые уже доступны, делает их слишком сложными. В таких случаях вместо того, чтобы вносить новые дополнения в существующие языки программирования, лучше начинать с нуля [7].


Некоторые из современных языков программирования, разработанные в 21 веке, приведены ниже [5]:


1.  **DART** - DART. JavaScript хорош для добавления базовых элементов интерактивности и создания веб-страниц, но программный код состоит из тысяч строк, что затрудняет использование языка. Вот почему этот язык программирования создан Google в качестве замены. Dart – новый язык, и он играет важную роль в устранении выявленных недостатков JavaScript. Оба данных языка используют аналогичный синтаксис и ключевые слова в языке C. Однако есть существенное отличие в том, что JavaScript основан на прототипах, в то время как Dart на классах и интерфейсах. Также Dart позволяет программистам добавлять переменные статического типа, такие как в C++ или Java.

2.  - *Ceylon*. Ceylon объединяет XML с Java и пытается решить проблемы в данной области.


3.  - Go [4]. Это универсальный язык программирования, поэтому он полезен для всех: его можно использовать как при разработке простых, так и сложных программ. В этом смысле он ближе к C и C++. Go включает в себя современные функции, удаляет ненужные данные из памяти и обеспечивает параллелизм.


4.  - *F#*. Он был разработан Microsoft, включает функциональность и удобство использования. Поскольку F# включен в состав языков программирования высокого уровня, он может поддерживать библиотеки и функции, а также другие языки CLR.


5.  - *Opa* не пытается заменить какой-либо конкретный язык. Он пытается заменить их все, организуя совершенно новую парадигму в веб-программировании. Его пользовательский интерфейс, база данных написаны на этом же языке.


6.  - *Fantom* был создан для кроссплатформенной переносимости. Проект Fantom не состоит только из компиляторов, он разделяет Java и .Net из

API и создает дополнительный уровень переносимости точно так же, как JVM или .Net выполняют байт-код для CLI.

7.  - Zimbu. Это уникальный и специфичный язык со своим собственным характером, в то же время, обладающий большим количеством особенностей. Он использует аналогичные выражения и операторы языку C, но ему принадлежат ключевые слова, типы данных и блочные структуры. Он свободно управляет памятью.

8.  - X10. Параллельное программирование в X10 было создано на основе модели разделенного глобального адресного пространства (PGAS). Коды и данные выбраны в блоках и разделены на разные «пространства».

9.  - NAHE. В настоящее время пользователи могут писать программы на NAHE, затем они могут компилировать программы в объектном коде на JavaScript, PHP, Flash / ActionScript или в байт-коде NekoVM. Готовятся дополнительные модули для компиляции в C# и Java.

10.  - Chapel является частью программы Cray Cascade, частично финансируемой и используемой американской администрацией на основе крупномасштабного проекта по экономически эффективным расчетам в области военной обороны (DARPA). Здесь используются алгоритмы параллельной обработки.

Различные популярные опросные агентства по всему миру с определённой периодичностью составляют рейтинговый список языков программирования. Существует множество рейтингов языков программирования [1]. Наиболее важным из них является индекс ТЮВЕ. На Рисунке 2 показана диаграмма с рейтингом языков программирования за 2022 год, согласно индексу ТЮВЕ.

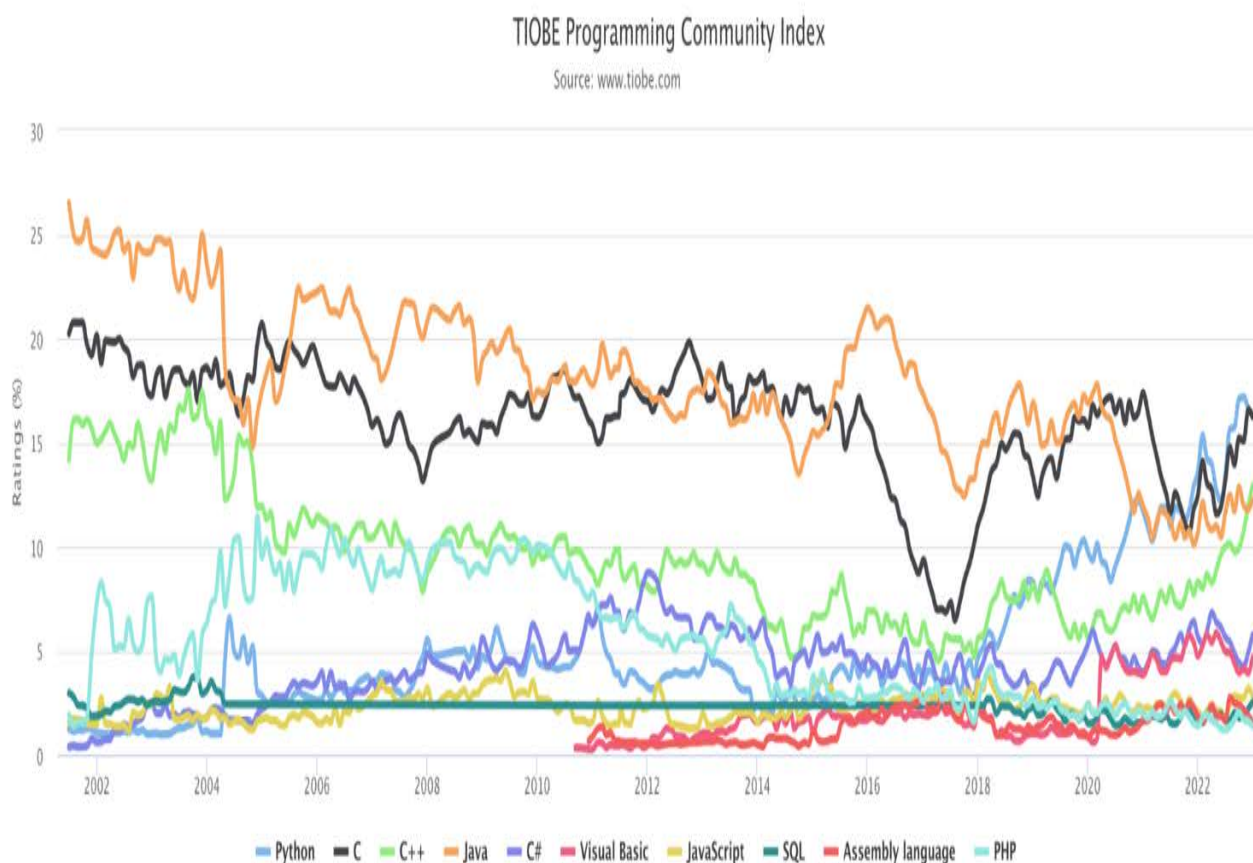


Рисунок 2. Рейтинг языков программирования по индексу TIOBE [8]

В число самых распространённых и широко используемых языков программирования включены Python [6], C, Java и C++. При этом не все эти языки являются современными, многие из них появились ещё в середине-конце 20 века, а потому они обладают рядом проблем и недостатков. Некоторые проблемы языков программирования показаны ниже:

1. Усложнение связи между различными системами и их компонентами из-за увеличения числа распределенных систем.
2. Проблема кроссплатформенности.
3. Проблемы типов системных переменных.
4. Проблемы с метаданными.
5. Для глобального программирования: проблемы наименования, обработки ошибок и безопасности.
6. Проблема управления версиями.

7. Масштабируемость. Распределенные системы могут отлично работать для сотен пользователей во внутренней сети организации, но проблемы могут возникнуть, когда десятки миллионов пользователей находятся в Сети:

- Проблемы с размером алгоритмов;
- Проблемы с системным программированием;
- Сложность современного программного обеспечения;
- Проверка и безопасность программ;
- Сложная структура микропроцессора;
- Проблемы с многоядерными процессорами.

Уровень языка характеризуется трудностью решения задачи с помощью этого языка. Чем проще написано решение задачи, тем более сложные операции и концепции реализуются, таким образом, объем программ уменьшается в зависимости от языка более высокого уровня. Современные этапы программирования нуждаются в тщательном изучении, и на основе полученных знаний могут быть реализованы новые методы разработки программного обеспечения. Поэтому производителям желательно использовать передовые методологические системы и иметь алгоритмическое мышление.

Библиографический список:

1. Голуб Л.Н., Довыденко О.В. Самые популярные языки программирования. // Научный журнал «Вестник образовательного консорциума Среднерусский университет. Информационные технологии». 2020. № 1 (15). С. 4-6. [Электронный ресурс]. URL: <https://www.elibrary.ru/item.asp?id=43029491> (Дата обращения: 21.04.2023).

2. Джумакулыев Д.О., Халмурадов М.Б. Языки программирования: типы и использование // Научный журнал «Вестник науки». 2023. Т. 1. № 3 (60). С. 277-281. [Электронный ресурс]. URL: <https://www.elibrary.ru/item.asp?id=50336433> (Дата обращения: 21.04.2023).

3. Ершов А.П., Покровский С.Б. Эволюция языков программирования // Научный журнал «Проблемы информатики». 2017. № 2 (35). С. 70-79. [Электронный ресурс]. URL: <https://www.elibrary.ru/item.asp?id=30362918> (Дата обращения: 25.04.2023).
4. Есилевский С. Языки программирования «новой волны». Язык GO. // Научный журнал «Системный администратор». 2021. № 7-8 (224-225). С. 65-73. [Электронный ресурс]. URL: <https://www.elibrary.ru/item.asp?id=46486850> (Дата обращения: 04.04.2023).
5. Иванов Д.П., Баев А.А., Современные языки программирования и сферы их применения // Научный журнал «Трибуна ученого». 2022. № 5. С. 70-80. [Электронный ресурс]. URL: <https://www.elibrary.ru/item.asp?id=49081322> (Дата обращения: 21.04.2023).
6. Никонорова Л.И., Тимофеев М.Г., Кузнецова А.П. Python как современный язык программирования // Научный журнал «Наука и Образование». 2019. Т. 2. № 2. С. 263-270. [Электронный ресурс]. URL: <https://www.elibrary.ru/item.asp?id=38578493> (Дата обращения: 25.04.2023).
7. Печенегина К. А. Выбор языков программирования: некоторые аспекты проблемы // Научный журнал «Молодежь. Общество. Современная наука, техника и инновации». 2020. № 19. С. 52-54. [Электронный ресурс]. URL: <https://www.elibrary.ru/item.asp?id=44709807> (Дата обращения: 25.04.2023).
8. Рейтинг ТЮБЕ за 2022 год. [Электронный ресурс]. URL: <https://www.tiobe.com/tiobe-index/> (Дата обращения: 04.04.2023).

Оригинальность 83%