

Муниципальное автономное общеобразовательное учреждение
"Средняя общеобразовательная школа № 7"
г. Боровичи Новгородской области

Проект по теме:

**«Оптимальный язык
программирования для начинающих»**

Выполнил: Скробат Дмитрий,
ученик 10А класса
Руководитель: Савельева Л.В.
учитель информатики

г. Боровичи

2022 год

Введение

Актуальность

В настоящее время в условиях открытой экономики, будь то отдельный человек, небольшое предприятие или крупная компания, вынуждены сверять свою деятельность с мировыми достижениями. Поэтому возникла необходимость в создании новой системы информационного обслуживания, основанной на современной технической базе и современных информационно-коммуникационных технологиях. Какую бы сферу жизнедеятельности человека мы не взяли: медицину, проектирование зданий, машин, образование, – без применения компьютерных технологий нигде в современном мире не обходится. Для каждой из этих областей разрабатываются соответствующие программы. Следовательно, сегодня является востребованной такая профессия, как программист: навыки программирования пользуются высоким спросом, должность программиста хорошо оплачивается. Даже за пределами IT-мира знание хотя бы одного языка программирования – это серьезный плюс в резюме.

Объект исследования

Объектом исследования являются современные языки программирования.

Предмет исследования

Предметом исследования являются характеристики и особенности языков программирования, которые учитываются при выборе их изучения.

Гипотеза

Я думаю, что сравнительные характеристики, предложенные в данном исследовании, помогут людям правильно выбрать язык программирования для изучения.

Цель

Основная цель данного исследования состоит в выявлении наиболее оптимального языка для начинающих программистов.

Задачи

Для достижения цели поставлены следующие задачи:

- 1) Перечислить области применения языков программирования
- 2) Обосновать выбор параметров для сравнения языков программирования.
- 3) На основании критериев выявить плюсы и минусы современных языков программирования
- 4) Разработать рекомендации для начинающих программистов по выбору изучения языка программирования.

Методы исследования

Теоретические:

- Анализ
- Сравнение
- *Систематизация*
- *Синтез*

Практические:

- Сравнение
- Анализ
- Синтез
- Систематизация

1. Теоретическая часть

1.1 Понятие языка программирования

Язык программирования — формальный язык, предназначенный для записи компьютерных программ. Язык программирования определяет набор лексических, синтаксических и семантических правил, определяющих

внешний вид программы и действия, которые выполнит исполнитель (обычно — ЭВМ) под её управлением.

Со времени создания первых программируемых машин человечество придумало более восьми тысяч языков программирования. Каждый год их число увеличивается. Некоторыми языками умеет пользоваться только небольшое число их собственных разработчиков, другие становятся известны миллионам людей. Профессиональные программисты могут владеть несколькими языками программирования.

Как правило, язык программирования определяется не только через спецификации стандарта языка, формально определяющие его синтаксис и семантику, но и через реализации стандарта — программные средства, обеспечивающих трансляцию или интерпретацию программ на этом языке; такие программные средства различаются по производителю, марке и варианту (версии), времени выпуска, полноте воплощения стандарта, дополнительным возможностям; могут иметь определённые ошибки или особенности воплощения, влияющие на практику использования языка или даже на его стандарт.

1.2 История создания и развития языков программирования

Первые программы писались на **машинном языке**, так как для ЭВМ того времени еще не существовало развитого программного обеспечения, а машинный язык — это единственный способ взаимодействия с аппаратным обеспечением компьютера, так называемым "железом".

Каждую команду машинного языка непосредственно выполняет то или иное электронное устройство. Данные и команды записывали в цифровом виде, например, в шестнадцатеричной или двоичной системах счисления. Человеку воспринимать программу на таком языке сложно. Кроме того, даже небольшая программа состояла из множества строк кода. Ситуация осложнялась еще и тем, что каждая вычислительная машина понимает лишь свой машинный язык.

Людам, в отличие от машин, более понятны слова, чем наборы цифр. Стремление человека оперировать словами, а не цифрами привело к появлению **ассемблеров**. Это языки, в которых вместо численного обозначения команд и областей памяти используются словесно-буквенные.

При этом появляется проблема: машина не в состоянии понимать слова. Необходим какой-нибудь переводчик на ее родной машинный язык. Поэтому, начиная со времен ассемблеров, под каждый язык программирования создаются *трансляторы* — специальные программы, преобразующие программный код с языка программирования в машинный код. Ассемблеры на сегодняшний день продолжают использоваться. В системном программировании с их помощью создаются низкоуровневые интерфейсы операционных систем, компоненты драйверов.

После ассемблеров наступил расцвет языков так называемого **высокого уровня**. Для них потребовалось разрабатывать более сложные трансляторы, так как языки высокого уровня куда больше удобны для человека, чем для вычислительной машины.

В отличие от ассемблеров, которые остаются привязанными к своим типам машин, языки высокого уровня *обладают переносимостью*. Это значит, что, написав один раз программу, программист без последующего редактирования может выполнить ее на любом компьютере, если на нем установлен соответствующий транслятор. Программа-транслятор для данной ЭВМ при трансляции исходного кода сама адаптирует его под эту ЭВМ.

Следующим значимым шагом было появление **объектно-ориентированных языков**, что в первую очередь связано с усложнением разрабатываемых программ. С помощью таких языков программист как бы управляет виртуальными объектами. Мыслить в рамках объектов-сущностей, описывать их взаимодействие, обобщать объекты в классы и устанавливать между ними наследственные связи, — все это делает программу по-своему похожей на реальный мир, на то, как его воспринимает человек.

На сегодняшний день в большинстве случаев реализация крупных проектов осуществляется с помощью объектно-ориентированных возможностей языков. Хотя существуют и другие современные парадигмы программирования, поддерживаемые другими или теми же языками.

1.3 Популярные языки программирования.

Чтобы выявить действительно самые востребованные на текущий момент времени языки в информационных технологиях, создаются различные рейтинги языков программирования. Одним из таких рейтингов является TIOBE.

Индекс TIOBE Programming Community — это показатель популярности языков программирования. Индекс обновляется раз в месяц. Рейтинги основаны на количестве квалифицированных инженеров по всему миру, курсах и сторонних поставщиках. Для расчета рейтингов используются популярные поисковые системы, такие как Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube и Baidu. Важно отметить, что индекс TIOBE не касается лучшего языка программирования или языка, на котором написано большинство строк кода.

Mar 2022	Mar 2021	Change	Programming Language	Ratings	Change
1	3	^	 Python	14.26%	+3.95%
2	1	v	 C	13.06%	-2.27%
3	2	v	 Java	11.19%	+0.74%
4	4		 C++	8.66%	+2.14%
5	5		 C#	5.92%	+0.95%
6	6		 Visual Basic	5.77%	+0.91%
7	7		 JavaScript	2.09%	-0.03%
8	8		 PHP	1.92%	-0.15%
9	9		 Assembly language	1.90%	-0.07%
10	10		 SQL	1.85%	-0.02%

В данной таблице представлены сравнительные характеристики на март 2021 г. и март 2022 г. В третьем столбике представлена смена позиций языка за один год. В столбце “Ratings” указаны результаты поисковых запросов, содержащих название языка. В последней колонке таблицы показаны изменения рейтинга языка за прошедший год.

Из данной таблицы мы наблюдаем, что за прошедший год язык программирования Python поднялся на первую строчку рейтинга TIOBE, а Java и C опустились на второе и третье место рейтинга соответственно. Самый большой рост за прошедший год показали языки Python и C++, прибавив 3.95% и 2.14% рейтинга соответственно, а самый низкий рост показал язык программирования C, потеряв 2.27% рейтинга.

Подобные списки публикуют многие компании, но все они по-разному подходят к вопросу определения степени популярности. Кто-то опрашивает самих разработчиков. Другие изучают размещаемые работодателями объявления о вакансиях, чтобы понять, какие навыки самые востребованные. А некоторые отталкиваются от числа поисковых запросов по тому или иному языку.

Таким образом, я предлагаю рассмотреть следующие языки программирования: Python, Java, C++, C#, JavaScript, PHP.

2. Практическая часть

2.1 Определение критериев анализа

Сравнивать языки программирования предлагаю по данным критериям:

- 1) Скорость работы конечного продукта;
- 2) Синтаксис языка программирования;
- 3) Кроссплатформенность;
- 4) Безопасность приложений.
- 5) Наличие и разнообразие фреймворков языка программирования;
- 6) Наличие актуальных областей применения;
- 7) Наличие учебных материалов.

Рассмотрим подробнее выбранные критерии.

К1: Скорость работы конечного продукта. Под скоростью работы программы, написанной на том или ином языке, подразумевается, насколько быстро работает конечный продукт.

К2: Синтаксис языка программирования — набор правил, описывающий комбинации символов алфавита, считающиеся правильно структурированной программой (документом) или её фрагментом.

К3: Кроссплатформенность - удобство и легкость запуска программы на разных платформах.

К4: Безопасное программирование — методика разработки программного обеспечения, предотвращающая случайное внедрение уязвимостей и обеспечивающая устойчивость к воздействию вредоносных программ и несанкционированному доступу. Баги и логические ошибки являются основной причиной появления уязвимостей программного обеспечения.

К5: Фреймворк —программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Он включает: вспомогательные программы, библиотеки кода, язык сценариев и другое программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. Чем больше фреймворков, тем больше можно решить задач с помощью языка программирования.

К6: В учебные материалы входят курсы, книги и видео по данному языку программирования, с помощью которых можно его изучить.

К7: Наличие актуальных областей применения – это направления программирования, которые являются наиболее востребованными. На сегодняшний день наиболее актуальны такие направления как искусственный интеллект, анализ данных, информационная безопасность, криптовалюты и блокчейн-технологии.

Теперь, когда критерии чётко обозначены, можно провести анализ выбранных языков программирования.

В начале обратимся к критерию 7 (**К7**). Для всех рассмотренных языков программирования существует огромное количество разнообразных курсов. Они могут быть полезны как для новичков в программировании, так и для более опытных программистов. Курсы можно найти платные и бесплатные, очные и заочные, с различным сроком обучения. С развитием дистанционного образования во многих высших учебных заведениях организованы курсы по изучению программирования в различных областях применения. Также при крупных IT компаниях существуют программы обучения, которые позволяют по их прохождению найти работу в данной компании.

Навыки владения компьютером и языком программирования можно получить, изучая специальную литературу, а также просматривая обучающие видеоролики на интернет платформах.

2.2 Python

К1: Python медленный — в среднем, на операции на нем понадобится в два, а то и в десять раз больше времени, чем, если бы вы выбрали другой язык. Для этого есть разные причины. Одна из них в том, что Python — язык с динамической типизацией. То есть на нем не нужно заранее определять тип данных, как в других языках. Конечно, это удобно разработчику, но такой подход требует большого резерва памяти для каждой переменной, чтобы она работала в любом случае. Соответственно, больше памяти означает больше времени на вычисления.

К2: Python похож на английский, поэтому его проще учить в сравнении с другими языками. По замыслу Python реализует преднамеренно простой и читабельный синтаксис, а также чрезвычайно согласованную модель программирования. На практике при чтении или написании кода программистам Python нет необходимости постоянно обращаться к справочному руководству.

К3: Python является кроссплатформенным языком программирования; это означает, что он работает во всех основных операционных системах. Любая программа на языке Python, должна выполняться на любом современном компьютере с установленной поддержкой Python. Например: перенос кода между Linux и Windows, как правило, сводится к копированию кода сценария между машинами.

К4: Было время, когда Python достигал пика с точки зрения уязвимостей, но сейчас он значительно улучшился. Согласно системе категорий аппаратных и программных слабостей и уязвимостей (Common Weakness Enumeration, CWE) в Python являются доминирующими следующие уязвимости: проверка входных данных (CWE-20, разрешения, привилегии и контроль доступа (CWE-264), утечка/раскрытие информации (CWE-200) и межсайтовый скриптинг (CWE-79). Самая низкая (в среднем 15%) уязвимость высокой степени серьезности за последние пять лет.

К5: Python существует так долго, что разработчики смогли сделать специальные библиотеки практически для любых целей. Например:

- NumPy - для многомерных массивов и высокоуровневых матриц.
- SciPy - для расчетов в инженерном деле подойдет.
- Pandas - для исследования, анализа и манипулирования данными.
- PyTorch — фреймворк машинного обучения с открытым исходным кодом, созданный на базе Torch. Используется для решения различных задач: компьютерное зрение, обработка естественного языка.

Если вам нужно решить какую-то вычислительную задачу, вероятно, что для нее уже есть специальная библиотека Python.

К6: Области применения: Web-разработка, анализ данных, машинное обучение, автоматизация процессов, системное программирование и администрирование.

2.3 PHP

К1: Язык PHP является интерпретируемым языком программирования. Это означает, что программа выполняется строка за строкой, а не

компилируется в исполняемый модуль. С одной стороны это приводит к тому, что программы выполняются более медленно по сравнению с компилируемыми языками, с другой стороны исходный код приложения может быть отредактирован в любой удобный момент.

К2: Язык программирования PHP довольно прост в изучении. Чтобы уловить логику его работы и понимать основной синтаксис, хватит 2-3 недель обучения; конечно, чтобы понять его «глубже», потребуется больше времени.

К3: PHP может работать на любых современных операционных системах (MacOS, Windows, Linux). А скрипты, написанные на этом языке, успешно функционируют на разном серверном ПО (Apache, IIS, Nginx и т. д.).

К4: PHP имеет наибольшее количество уязвимостей среди всех языков. Это единственный язык с уязвимостью SQL Injection (CWE-89), которая росла в 2017 и 2018 годах. Другая распространенная уязвимость, связанная с PHP, - межсайтовый скриптинг (CWE-79). Средняя уязвимость высокой степени серьезности за последние пять лет составляет 16%.

К5: *CodeIgniter* — это фреймворк PHP, использующий архитектуру Model View Controller (MVC). С точки зрения новичка, это означает, что CodeIgniter использует различные компоненты для решения конкретных задач разработки. Этот подход популярен среди разработчиков, потому что он позволяет создавать масштабируемые веб-приложения с меньшими размерами.

Laravel — это бесплатный опенсорсный PHP-фреймворк, который появился в 2011 году. С тех пор он смог стать тем фреймворком, который чаще всего выбирают веб-разработчики. Одна из главных причин этого заключается в том, что Laravel позволяет разрабатывать сложные веб-приложения проще, быстрее и безопаснее, чем любые другие фреймворки.

Yii — это простой, но высокопроизводительный универсальный фреймворк, основанный на компонентной структуре. Он известен благодаря

высокой производительности, но, в первую очередь, он знаменит своей простотой. Этот фреймворк появился в декабре 2008 года. Он позволяет использовать код сторонних разработчиков, а имеющийся в нём генератор кода Gii позволяет быстро создавать базовые структуры, на основе которых можно строить собственные решения.

К6: Области применения: разработка веб-приложений, работа с базами данных, создание GUI-приложений.

2.4 C#

К1: Когда говорят C#, нередко имеют в виду технологии платформы .NET (Windows Forms, WPF, ASP.NET, Xamarin). И, наоборот, когда говорят .NET, нередко имеют в виду C#. Согласно ряду тестов веб-приложения на .NET 6 в ряде категорий сильно опережают веб-приложения, построенные с помощью других технологий. Приложения на .NET 6 в принципе отличаются высокой производительностью.

К2: C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Понятный синтаксис C# заметно упрощает не только разработку как таковую, но и другие важные аспекты совместной работы, например, чтение чужого кода. Это упрощает процесс рефакторинга и исправления ошибок при работе над приложениями в больших командах.

К3: Как и Java, C# также является нейтральным или независимым от платформы языком, код которого может быть скомпилирован и запущен во всех операционных системах. Он обычно используется с платформой Microsoft .NET для Windows.

К4: У общезыковой исполняющей среды (common language runtime, CLR) в .NET Framework есть своя модель безопасного выполнения кода, независимая от ограничений операционной системы, в которой она работает. Более того, в отличие от старой модели защиты на основе участников безопасности (principal-based security), CLR реализует политику, исходя из того, откуда поступает код, а не из того, кто является его пользователем. Эта

модель защиты по правам доступа кода (code access security) имеет больший смысл в современных условиях, поскольку немалая часть кода устанавливается через интернет, и даже доверенный пользователь (trusted user) не знает, какой код действительно безопасен.

K5: *Microsoft .NET Compact Framework (.NET CF)* — это версия .NET Framework, которая разработана для запуска приложений на устройствах, основанных на платформе Windows CE, таких как PDA, мобильные телефоны, заводские контроллеры и других. .NET Compact Framework использует библиотеки, подобные полной версии .NET Framework, вместе с тем имеются специальные наборы библиотек для мобильных устройств, например, Windows CE InputPanel. Разработка приложений, использующих возможности .NET Compact Framework может производиться в Visual Studio .NET 2003, в Visual Studio 2005 и Visual Studio 2008, на языке C# или Visual Basic .NET. Результатом разработки являются специальные приложения для мобильных устройств, обладающие высокой производительностью (исполняются при помощи JIT компилятора).

Microsoft Silverlight — это программная платформа для написания и запуска многофункциональных интернет-приложений RIA, схожая с Adobe Flash, включающая в себя модуль для браузера, который позволяет запускать приложения, содержащие анимацию, векторную графику и аудио-видео ролики, что характерно для RIA. Версия 2.0 добавила поддержку для языков .NET и интеграцию с IDE.

.NET Framework — программная платформа, выпущенная компанией Microsoft в 2002 году. Основой платформы является общезыковая среда исполнения Common Language Runtime (CLR), которая подходит для различных языков программирования. Функциональные возможности CLR доступны в любых языках программирования, использующих эту среду. В настоящее время .NET Framework развивается в виде .NET.

К6: Области применения: разработка веб-приложений, разработка мобильных игр, разработка игр на ПК, разработка программного обеспечения для защиты систем, разработка приложений для Windows.

2.5 Java

К1: Любой высокоуровневый язык из-за использования виртуальной машины для своей компиляции обладает более низкой производительностью. Java в том числе, но не только это влияет на ее «быстроту». Также производительность падает из-за проблем с очисткой памяти, плохой настройки кэширования, вероятности блокировки потоков и т. д.

К2: При разработке Java было уделено большое внимание простоте языка, поэтому программы на Java, по сравнению с программами на других языках, проще писать, компилировать, отлаживать и изучать. Язык Java очень похож на языки C и C++. От C язык Java унаследовал синтаксис, а от C++ — объектную модель.

К3: Java переносим с точки зрения разработчика: код, написанный на Java, может быть выполнен в любой среде без необходимости перекомпиляции.

К4: Важной особенностью технологии Java является гибкая система безопасности благодаря тому, что исполнение программы полностью контролируется виртуальной машиной. Любые операции, которые превышают установленные полномочия программы (например, попытка несанкционированного доступа к данным или соединения с другим компьютером) вызывают немедленное прерывание.

К5: *Spring Framework* — одна из самых мощных и широко используемых платформ Java. Благодаря своей концепции внедрения зависимостей и особенностям аспектно-ориентированного программирования Spring покорила мир разработки. Это платформа с открытым исходным кодом, используемая для корпоративных приложений. Spring чаще всего используют в разработке веб-приложений. Его применяет ряд технологических гигантов, включая Netflix и Amazon. Главное

преимущество фреймворка заключается в том, что он чрезвычайно лёгкий и не требует вызова веб-сервера.

Dropwizard — мощный Java-фреймворк, высоко оптимизированный под разработку RESTful-сервисов. Он также идеален для написания микросервисов на этом языке и даёт лёгкий доступ ко всем мощным Java-библиотекам, например Google Guava, Jetty server, Hibernate Validator, Logback, Joda Time, а также Jackson для обработки JSON-файлов. JSON избавляет от необходимости писать код для метрик и конфигураций, позволяя направить силы на функционал приложения.

MyBatis фреймворк для осуществления маппинга между Java-приложениями и базами данных SQL. Обычно для подключения приложения к реляционной БД необходим API Java Database Connectivity. Он позволяет разработчикам выполнять крупные SQL-операции за несколько строчек кода.

К6: Области применения языка: разработка веб-приложений, разработка мобильных приложений, разработка игр, работа с большими данными, разработка серверных приложений.

2.6 C++

К1: Грамотно написанные программы на C++ могут работать просто молниеносно. Одной из основных проектных характеристик языка C++ была высокая производительность. Если же в вашей программе требуется добиться высокой производительности, то C++ позволяет работать и с ассемблером. Ассемблер - это самый низкоуровневый человекочитаемый язык программирования, взаимодействующий непосредственно с аппаратным обеспечением компьютера.

К2: У C++ много разных функций и относительно сложный синтаксис. Код на этом языке писать не так просто.

К3: C++ является компилируемым языком, а это значит, что компилятор транслирует исходный код на C++ в исполняемый файл, который содержит набор машинных инструкций. Но разные платформы имеют свои особенности, поэтому скомпилированные программы нельзя просто

перенести с одной платформы на другую и там уже запустить. Однако на уровне исходного кода программы на C++ по большей степени обладают переносимостью, если не используются какие-то специфичные для текущей ОС функции. А наличие компиляторов, библиотек и инструментов разработки почти под все распространенные платформы позволяет компилировать один и тот же исходный код на C++ в приложения под эти платформы.

К4: Неоправданное обилие побочных эффектов в сочетании с отсутствием контроля со стороны системы времени исполнения языка и слабой системой типов делает программы на C++ подверженными непредсказуемым фатальным сбоям (общезвестные падения с сообщениями типа «Access violation», «Pure virtual function call» или «Программа выполнила недопустимую операцию и будет закрыта»), что исключает применение C++ при высоких требованиях к отказоустойчивости. Кроме того, это увеличивает длительность самого процесса разработки.

К5: *Poco* - набор open-source библиотек для создания кроссплатформенных сетевых и веб-приложений на C++, хорош тем, что множество типовых задач в библиотеке уже решены, что уменьшает время разработки и ошибки, плюс реализована удобная работа с базами данных SQL, MonoDB и Redis.

SDL- простая, удобная, кроссплатформенная библиотека как для создания 2D игр, так и для пользовательских приложений, имеются инструменты для профессиональной работы со звуком.

Qt — очень мощный набор инструментов для создания кроссплатформенных приложений на языке C++ и Python. Qt уже достаточно старый фреймворк, но он продолжает активно развиваться и на нем написаны такие программы как: 2ГИС для Android, Kaspersky Internet Security, Virtual Box, Skype, VLC Media Player, Opera и другие.

К6: Области применения: разработка игр, разработка настольных и кроссплатформенных приложений, разработка операционных систем, мобильная разработка, разработка веб-приложений.

2.7 JavaScript

К1: Преимущество скорости JavaScript получает благодаря тому, что код не компилируется на стороне клиента, ведь скрипт выполняется в браузере пользователя. Это значительно уменьшает нагрузку на сервер по сравнению с веб-приложениями, написанными на других языках.

К2: Синтаксис языка JavaScript во многом напоминает синтаксис C и Java. Сначала код может показаться сложным, но к его синтаксису, логике быстро привыкаешь. Отображение действий ощутимо добавляет энтузиазма.

К3: Скрипты, написанные на JavaScript, поддерживаются всеми браузерами, операционными системами и платформами.

К4: В интернете легко можно найти и скачать исходные коды для популярных скриптов. Они находятся в свободном доступе. Поэтому сложно гарантировать, что веб-приложение на JavaScript не будет взломано. Ухудшает ситуацию и то, что скрипт выполняется на стороне пользователя каждый раз во время загрузки страницы, а не на сервере, где можно было бы отследить поломку кода.

К5: *jQuery*, пожалуй, самая популярная библиотека JavaScript с таким количеством функций для современной разработки. *jQuery* — это быстрая и лаконичная библиотека JavaScript, созданная Джоном Резигом в 2006 году. Это кроссплатформенная библиотека JavaScript, предназначенная для упрощения HTML-скриптинга на стороне клиента. Более 19 миллионов веб-сайтов в настоящее время используют *jQuery*! Такие компании, как WordPress, Facebook, Google, IBM и многие другие, полагаются на *jQuery* для обеспечения своего рода просмотра веб-страниц.

Vue — это JavaScript-фреймворк с открытым исходным кодом для создания креативного интерфейса. Интеграция с *Vue* в проектах, использующих другие библиотеки JavaScript, упрощена, поскольку она разработана для адаптации. Более 36 000 веб-сайтов в настоящее время используют *Vue*. Такие компании, как Stackoverflow, PlayStation и т.д., полагаются на *Vue* для своих сайтов пользовательского интерфейса.

Angular — одна из самых мощных сред JavaScript. Google использует эту платформу для разработки одностраничного приложения (SPA). Эта среда разработки известна прежде всего потому, что она предоставляет разработчикам лучшие условия для объединения JavaScript с HTML и CSS. Более полумиллиона сайтов, таких как Google, Youtube и т.д., используют Angular.

К6: Области применения: разработка веб-приложений, разработка браузерных операционных систем, разработка серверных приложений, разработка мобильных приложений, разработка прикладного программного обеспечения.

2.8 Анализ языков программирования

Из выше изложенного можно составить сравнительную таблицу.

Язык программирования	Возможности	Преимущества	Недостатки
Python	Анализ данных, машинное обучение, автоматизация процессов, системное программирование, разработка веб-приложений.	Универсальность, кроссплатформенность, понятный синтаксис, большое количество фреймворков.	Низкая производительность.
PHP	Разработка веб-приложений, работа с базами данных, создание GUI-приложений.	Понятный синтаксис, кроссплатформенность, легкость редактирования кода.	Большое количество уязвимостей, узкая специализация.
C#	Разработка веб-приложений, мобильных игр, игр на ПК, программного обеспечения для защиты систем.	Высокая производительность, безопасность, наличие большого количества библиотек.	Зависимость от платформы .NET.
Java	Разработка веб-приложений, мобильных приложений, серверных приложений, работа с большими данными.	Кроссплатформенность, безопасность, понятный синтаксис, большое количество фреймворков.	Низкая производительность.
C++	Разработка игр, настольных и кроссплатформенных приложений, операционных систем.	Высокая производительность, возможность работать с данными на низком уровне.	Сложный синтаксис, сложно писать кроссплатформенный код.

JavaScript	Разработка веб-приложений, прикладного программного обеспечения, разработка браузерных операционных систем.	Понятный синтаксис, высокая производительность, кроссплатформенность, большое количество фреймворков.	Большое количество уязвимостей, узкая специализация.
------------	---	---	--

Заключение и выводы

Таким образом, выбирая первый язык программирования, стоит оценить следующие факторы: что именно вы сможете создать в процессе изучения, сложность изучения языка, перспективы языка и востребованность его на рынке труда. Чтобы выбрать язык программирования, определитесь, какие задачи вы хотите решать. Оцените, какие дополнительные навыки потребуются, какие личностные черты важны. Если вы еще до конца не определились, чем именно хотите заниматься, начните изучать язык программирования Python.

Рекомендация Python объясняется простотой его освоения и наличием достаточно мощных универсальных средств, пригодных для решения научных, технических и экономических задач, а также задач бытового характера, игровых и т.д. Вы встретите Python в разработке веб-приложений, в машинном обучении, в анализе данных и в системном программировании. Python используется не только в крупных компаниях, но в организациях любого масштаба и бизнеса. Он предлагает большую стандартную библиотеку, возможность работы программ на разных платформах и полезный функционал управления памятью.

Для изучения основ языка Python можно воспользоваться образовательной платформой Stepik, на которой предлагаются разнообразные онлайн курсы. Я рекомендую "Поколение Python": курс для начинающих. Курс посвящен основам программирования на языке Python. Он состоит из 8 модулей. В конце некоторых модулей вас ожидают самостоятельные и контрольные работы с ограничением времени

на их выполнение. Они помогут понять, насколько хорошо вы усвоили тему. Курс вы можете изучать в удобное для вас время.

Если вы уже знаете, какие программы хотите разрабатывать, то используйте таблицу для выбора первого языка программирования.

Источники информации

- 1) C++ для начинающих. Шаг за шагом. Шилдт Герберт, ЭКОМ Паблишерз, 2013.
- 2) A Byte of Python. Swaroop Chitlur
- 3) Изучаем Python. Программирование игр, визуализация данных, веб-приложения. Мэтис Эрик. Издательский дом «Питер», 2019.
- 4) Изучаем Python. 5-е изд. Том 1. Марк Лутц. Вильямс, 2019
- 5) <https://www.tiobe.com/tiobe-index/>