**Системное программирование**

* **Системное программирование** – раздел программирования, в котором сочетаются исследования новых архитектур, алгоритмов, структур данных и др. и деятельность по проектированию, разработке, тестированию и сопровождению (поддержке) системного программного обеспечения (системного ПО), т. е. для создания новых информационных технологий.

Оптимальной средой для системного программирования является Visual Studio IDE, так как Visual Studio — это мощное средство разработчика, которое можно использовать для выполнения всего цикла разработки в одном месте. Это комплексная интегрированная среда разработки (IDE), которую можно использовать для записи, редактирования, отладки и сборки кода, а затем развертывания приложения. Помимо редактирования и отладки кода Visual Studio включает компиляторы, средства завершения кода, управление версиями, расширения и многое другое, чтобы улучшить каждый этап процесса разработки программного обеспечения.

Для системного программирования хорошо подойдет язык С++. Из его преимуществ можно выделить следующее:

1. Поддержка объектно-ориентированного программирования (ООП). ООП помогает сделать код проще, и его быстрее писать.
2. Высокая скорость.
3. Возможности для работы с данными на низком уровне — то есть на уровне, близком к аппаратному. Благодаря этому на С++ можно писать драйвера, микроконтроллеры.

* **Сетевое программирование** – это процесс разработки программного обеспечения, которое взаимодействует с другими компьютерами или устройствами через сетевые соединения.

Для этих задач подойдет среда Node.js, так как Node.js обеспечивает возможность разработки приложений с использованием двух распространенных языков программирования — JavaScript и TypeScript. А это огромное преимущество, учитывая высокую популярность и востребованность этих языков на рынке труда.

Совместимость с JavaScript и TypeScript делает процесс разработки более удобным и доступным, позволяя разработчикам использовать в новых проектах уже сформированные навыки и опыт работы. Node.js имеет огромное количество встроенных модулей, а также большое количество сторонних библиотек, широко используется для создания веб-серверов. Благодаря асинхронной обработке и архитектуре V8, Node.js способен обрабатывать большое количество запросов с минимальной задержкой, что делает его хорошим выбором для создания высоконагруженных приложений. Кроме того, Node.js позволяет легко создавать API-серверы, которые предоставляют данные клиентам или другим приложениям.

Работая с Node.js, стоит использовать язык JavaScript, так как он является основным для этой платформы, следовательно, его применение делает процесс разработки максимально гибким и минимально затратным по времени.

* **База данных** – это набор данных, который как-то структурирован.

Для работы с базами данных подходит JetBrains Data Grip. JetBrains DataGrip — это мощный инструмент для работы с базами данных. Он поддерживает самые разные базы данных, включая PostgreSQL, MySQL, SQL Server, Oracle, MongoDB, а также источники данных DDL. С помощью проводника баз данных можно легко просматривать структуру базы данных и выполнять другие действия. DataGrip имеет настраиваемый интерфейс, подсветку синтаксиса, автодополнение кода и форматирование SQL. Он также включает в себя функции для анализа и отладки данных. Оптимальным языком для работы с базами данных является SQL, так как табличная структура его понятна, а сам язык просто изучить. Благодаря обеспечению различного представления данных, можно сделать так, чтобы тот или иной пользователь видел различные представления таблицы. Кроме того, SQL поддерживает архитектуру «клиент-сервер», позволяет манипулировать БД, обеспечивая возможность динамического изменения и расширения структуры, поддерживает возможность программного доступа к БД.

* **Работа и программирование систем искусственного интеллекта**: лучше всего для программирования систем ИИ подходит среда TeamsFlow. Это открытая платформа для глубокого обучения. Она предоставляет широкий набор инструментов и библиотек для разработки различных моделей машинного обучения и нейросетей. Его большое сообщество пользователей и разработчиков, что делает его очень популярным и доступным для обмена знаниями и опытом. Основным языком для работы с TeamsFlow является Python. Python имеет обширную экосистему библиотек и инструментов, которые могут быть использованы в совокупности с TensorFlow для разработки более сложных моделей и решений в области машинного обучения и искусственного интеллекта, а также обладает удобным и доступным синтаксисом.
* **Работа с визуальными средами, дополненная реальность, различные варианты графической работы.** Для работы с визуальными средами самой оптимальной платформой может быть Unity. Основными преимуществами Unity являются наличие визуальной среды разработки, межплатформенной поддержки и модульной системы компонентов. Unity обладает множеством встроенных инструментов и функций, таких как физическая симуляция, анимация, управление ресурсами и другие. Это позволяет разработчикам создавать разнообразные и интерактивные игровые механики и эффекты. К недостаткам относят появление сложностей при работе с многокомпонентными схемами и затруднения при подключении внешних библиотек. На Unity написаны тысячи игр, приложений и симуляций, которые охватывают множество платформ и жанров. При этом Unity используется как крупными разработчиками, так и независимыми студиями. Основным языком для работы с Unity является C#, который широко распространен и удобен. Реже используется такой язык, как Unity Script, разработанный специально для использования в Unity.