

Глоссарий основных терминов курса

1. *Алгоритм (Algorithm)*. Шаги или инструкции, которые описывают решение конкретной задачи или операцию.
2. *Багфиксинг (bug fixing)* – это процесс устранения неисправностей в ПО. Под неисправностью следует считать несоответствие поведения программы техническому заданию.
3. *Джун* (от англ. junior, младший) – работает над простыми задачами под руководством опытных коллег.
4. *Интерпретатор (Interpreter)*. Программа, которая выполняет и интерпретирует код программы построчно во время выполнения.
5. *Компилятор (Compiler)*. Программа, которая преобразует весь исходный код программы в исполняемый машинный код до его выполнения.
6. *Костыль* – решение, которое сделано для устранения локальной проблемы в коде. Оно работает, но не решает проблемы в целом.
7. *Легаси* – с английского legacy переводится как «наследие». Легаси-код – это код, который перешёл «по наследству» от предыдущих разработчиков.
8. *Либа* – lib, library, библиотека.
9. *Мидл* (от англ. middle, средний) – может решать сложные задачи или писать код самостоятельно, старшие коллеги проверяют только результаты.
10. *Отладка (Debugging)*. Процесс поиска и исправления ошибок в программе. Отладка помогает идентифицировать проблемные места в коде и устранить ошибки.
11. *ООП (Object-Oriented Programming) (объектно-ориентированное программирование)* - методология программирования, в которой код организуется в объекты, которые взаимодействуют друг с другом через методы и свойства.
12. *Релиз (release)* – готовая программа.
13. *Сеньор* (от англ. senior, старший) – всё делает сам: пишет код, занимается архитектурой, обучает младших коллег. Решает сложные задачи и отвечает за результаты.
14. *Технологический стек (stack)* – сложная комбинация, включающая языки программирования, программное обеспечение и спектр фреймворков, применяемых для разработки IT-проекта.
15. *Фича* – это сленговое слово, которое пришло в русский язык из компьютерного и в переводе с английского означает «особенность» (feature). Это не просто отличительная черта, а именно полезная дополнительная функция, классная доработка или привлекательная деталь гаджета, улучшенная характеристика продукта.
16. *API (Application Programming Interface)*. Набор инструкций и правил, которые определяют, как различные программы и компоненты программного обеспечения могут взаимодействовать между собой.

17. *Backend-разработка* – это набор аппаратно-программных средств, при помощи которых реализована логика работы сайта. Попросту говоря, это то, что скрыто от глаз пользователя и происходит вне его браузера и компьютера.

18. *Frontend -разработка* – это создание пользовательского интерфейса на клиентской стороне веб-сайта или приложения. Это всё, что видит пользователь, когда открывает веб-страницу, и с чем он взаимодействует: кнопки, баннеры и анимация

19. *Framework (фреймворк)* – это программная платформа, которая упрощает разработку программного продукта, определяет структуру проекта и помогает удобно объединять в нём разные компоненты. Пример: Unity 3D, Django, Google.Colab, React.

20. *Git*. Распределенная система контроля версий, используемая для отслеживания изменений в коде и совместной работы над проектами.

21. *IDE (Integrated Development Environment)*. Интегрированная среда разработки, которая предоставляет программисту удобную среду для написания, отладки и выполнения программного кода.

22. *PIP (Python Package Installer)*. Инструмент командной строки, используемый для установки и управления пакетами Python. PIP упрощает процесс установки сторонних библиотек и модулей.

23. *Package (пакет)*. Способ организации и структурирования модулей Python. Пакеты позволяют группировать связанный функционал вместе для удобного использования.