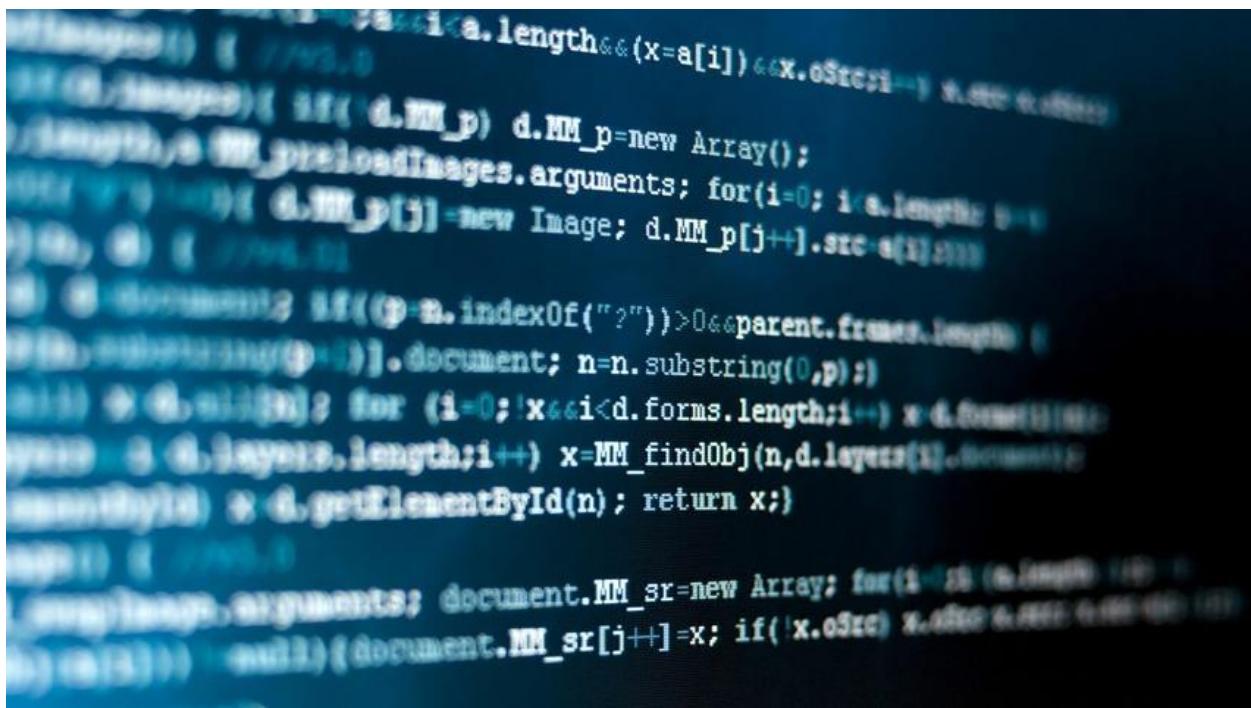


Наръчник за начинаещи: програмиране и софтуерни технологии

Все по-често чуваме, че програмирането и ИТ познанията са все по-важни, търсени и перспективни. Казват, че има много свободни работни позиции в софтуерната индустрия, както и опции за растеж и развитие. Какво обаче представлява програмирането, разработката на софтуер, кои са най-популярните програмни езици днес, ще се опитаме да разнищим тук.



Що е то програмиране?

Същността на **програмирането** е да се управлява работата на компютъра на различните му нива. Управлението става с помощта на заповеди (команди) от програмиста към компютъра. Командите се издават в писмен вид и биват безпрекословно изпълнявани от компютъра. Те са много на брой и за издаването им се използват различни видове програмни езици.

Езици за програмиране

Чрез **програмните езици** програмистите подават инструкции за изпълнение на компютъра. Програмните езици могат да се използват за създаване на програми, които контролират поведението на компютъра или пък реализират алгоритми.

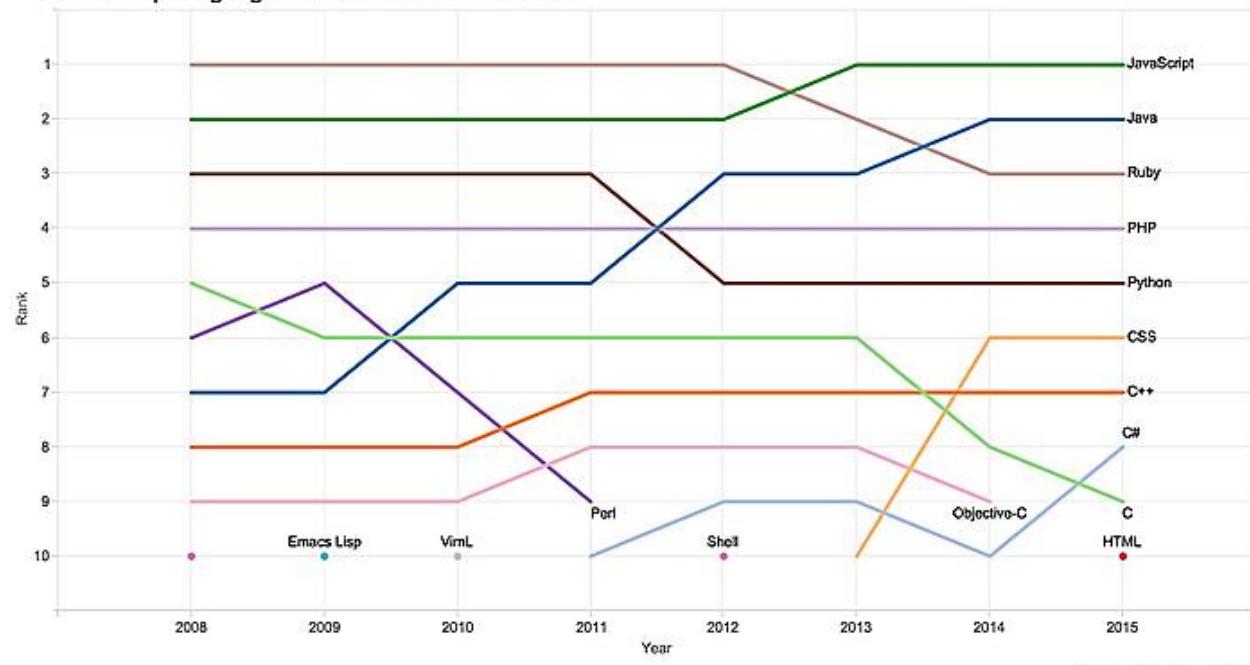
Първите програмни езици въобще предхождат изобретяването на компютъра и са били използвани за управление на поведението на различни машини, като например механично пиано. Хиляди програмни езици са създадени и продължават да се създават всяка година. Някои от тях стават широко използвани. Такива примери са: C, C#, Java, C++, PHP, JavaScript и т.н.

Повечето програмни езици извършват изчисления в императивен вид (т.е. като последователност от команди за изпълнение). Някои езици използват други форми на спецификация като декларативната форма (т.е. задава се желания резултат, а не как да се постигне). Такива са езиците, които поддържат функционално или логическо програмиране.



Най-популярните програмни езици днес

Rank of top languages on GitHub.com over time



Source: GitHub.com

Трудно е да се определи точната подредба на най-популярните езици за програмиране днес, защото според [GitHub JavaScript води класацията](#), а според [Inc](#) пък [начело застава Java](#). Общо взето, сред най-разпространените програмни езици се нареждат: Java, C#, C/C++, PHP, JavaScript, Python, Ruby, Objective-C/Swift. HTML и CSS са също изключително често ползвани, но те не са точно езици за програмиране, а по-скоро езици за форматиране на текст и презентация на документи (най-просто казано).

Голяма част от тези езици се изучават в [програмата на СофтУни](#). Важното е да се схване процеса при разбиване на един проблем на малки подзадачи и последователността от писането на кода за решението. Веднъж щом я има тази основа, добрият програмист може да превключва между езиците, тъй като синтаксисът им е доста подобен.

Ето и кратко описание на възможностите на най-популярните езици:

- **C** – Оригиналният език C, който е създаден през 1972 г., е все така разпространен и днес. Той не само работи на почти всички възможни компютърни платформи, а е и изключително стабилен. С е предшественик на C# и Java и е идеален за работа с програми на ниско ниво (например за програмиране на телевизори, операционни системи на самолети, както и за операционната система на Microsoft – Windows).
- **Java** – Изначално, когато е създаден през 1991 г., програмният език на Oracle – Java, е бил предназначен за програмиране на смарт телевизори. Днес обаче той се използва за разработка на Android приложения, като самата мобилна операционна система е разработена на Java. Под „Java“ обикновено се подразбира не само езикът Java, но и платформите Java и Java EE. На този език се пишат както сложни server-side системи, така и уеб базиран софтуер, а дори и големи индустриални системи, включващи сложна обработка и изискващи голяма надеждност.
- **C#** - Този програмен език, произнасян „си шарп“, е друга вариация на C. Създаден е от Microsoft и въплъщава в себе си доста от концепциите на Java. Езикът C# се ползва за изграждане на всякакви софтуерни приложения и системи: от тежка сървърна логика (където конкурира Java) до уеб приложения (където конкурира PHP) и настолни приложения (където е почти без конкуренция върху Windows платформата), както и мобилни приложения (където конкурира Java за Android устройствата и Objective-C за iOS устройства). C# е много подходящ за начинаещи, ето защо и е въвеждащ в програмата на СофтУни.
- **JavaScript** – JavaScript е скриптов език, който се използва главно за изграждане на потребителски интерфейс в уеб и мобилни приложения. На него се пише клиентската част на приложенията (това, което вижда потребителят) и рядко се използва за сървърната част (логиката, съхранението и обработката на данните). Технологията Node.js позволява JavaScript да се използва за сървърни приложения, но по една или друга причина тя не се използва масово. JavaScript няма нищо общо с Java, освен приликата в името. ☺
- **PHP** – PHP е един от най-често срещаните езици в уеб пространството – според някои, той обхваща 1/3 от всички уеб сайтове. Използват го големи сайтове като WordPress, Facebook, Yahoo. PHP е език за създаване на динамични уеб сайтове и клиентски уеб приложения. PHP (почти) винаги се ползва заедно с HTML, CSS и JavaScript.
- **C++** – Използва се предимно за готови софтуерни продукти, като например игри, офис приложения, графични и видео редактори, операционни системи.
- **Objective-C и Swift** – Objective-C доскоро беше основният програмен език, използван от Apple за операционните им системи OS X и iOS. От юни 2014 г. обаче, когато Apple представиха собственият им програмен език Swift, разработката на Apple софтуер и приложения малко по малко се поема от Swift. Swift е създаден с идеята да бъде по-устойчив на грешен код (т. е. по-безопасен) от Objective-C, а също и сбит (т.е. да пишем по-малко код).

С кой програмен език да започна?



Много хора питат с кой език за програмиране да започнат. Това е сложен въпрос и няма единозначен отговор. Зависи от целта им. Например, ако някой иска да се научи да създава уеб сайтове, най-добре е да започне от HTML + CSS + JavaScript и да продължи с PHP, бази данни и уеб услуги. Ако някой иска да създава уеб приложения, може да започне от HTML + CSS + JavaScript и паралелно да учи програмиране със C# или Java и бази данни. Програмирането пък включва в себе си структури от данни, алгоритми, обектно-ориентирано програмиране и качествен програмен код. Ако някои иска да учи разработка на мобилни приложения за iOS, нещата са съвсем различни и трябва да учи Objective-C/Swift + iOS платформата.

Нашият съвет е да се научи поне един език с общо предназначение (C# или Java), за да се разберат основните концепции на програмирането. Добра идея е да включите и поне един скриптов език (като JavaScript или PHP). Към това добавете и основни технологии в разработката на софтуер като бази данни, уеб услуги, структури от данни и алгоритми и вече имате една добра основа.

Какво е front-end и back-end разработка? Каква е разликата?

Както вече видяхме, софтуерните технологии и програмните езици се използват за различни цели и за създаването на различни видове софтуер. Най-общо можем да ги обединим в следните категории:

- Технологии за **потребителски интерфейс (front-end)**: например HTML, CSS, JavaScript;
- Технологии за **сървърно програмиране (server-side)**: C#, Java, PHP, Python;

- Технологии за **работка с данни (back-end)**: бази данни и SQL, ORM технологии, уеб услуги и REST;
- Други технологии и платформи: **мобилни платформи**, cloud технологии, софтуерно инженерство, операционни системи, вградени системи, езици от по-ниско ниво като C и C++

В СофтУни залагаме сериозно и на **front-end**, и на **back-end технологии** – във второто ниво на обучение – Advanced, има [по цял модул от курсове](#) в съответната област.

Етапи в създаването на софтуер

Както при всичко друго, и при създаването на софтуер има определен процес, който се следва.

Етапите са следните:

- **Изисквания** – определя се целта на софтуера, който предстои да бъде разработен, различните спецификации и изисквания;
- **Дизайн** – тук се има предвид предимно планиране на софтуерното решение, както и архитектура на кода;
- **Писане на програмен код** – подразбира се създаването на софтуера и основна проверка на качеството;
- **Тестване** – компонентно тестване (unit testing) и интеграционно тестване или изпълнение на програмата (приложението) с цел намиране на софтуерни бъгове;
- **Дебъгване** – методически процес по намиране и намаляване броя на софтуерни бъгове и грешки;
- **Деплоймънт** – внедряване на вече готовото софтуерното решение, за да може да бъде използвано според първоначалните цели и идеи;
- **Поддръжка** – след като софтуерът е внедрен, важно е той да бъде поддържан – да се отстраняват появили се грешки и проблеми, да се оптимизира и подобрява.



Източници и ресурси за учене на програмиране

Само посещението на учебните занятия е крайно недостатъчно, за да навлезете и напреднете в такава динамична професия, каквато е тази на програмиста. Ето защо ние препоръчваме да използвате и външни ресурси за учене:

- [Всички материали](#) по курсовете, които организира СофтУни, са достъпни на сайта ни
- „[Въведение в програмирането със C#](#)“, автори Светлин Наков и колектив

- „[Въведение в програмирането с Java](#)“, автори Светлин Наков и колектив
- „[Fundamentals of computer programming with C#](#)“ – английската версия на книгата на Наков
- [Stack Overflow](#) – един от най-посещаваните сайтове от всички програмисти, там може да намерите отговори на почти всичките си въпроси (а ако пък не успеете, можете направо да зададете въпроса си във форума)
- [Codecademy](#) – интерактивен сайт за бесплатно основно обучение по различни технологии
- [Computer Programming в Quora](#) – във форума често има отговори на различни въпроси относно „Какво е програмиране“, „Защо да се занимавам с програмиране“, „Коя книга е най-добра за програмния език X“ и т.н.

