# Предговор

Ако искате да се захванете сериозно с програмиране, **попаднали сте на правилната книга**. Наистина! Това е книгата, с която можете да направите първите си стъпки в програмирането. Тя ще ви даде солидни основи от знания, с които да поемете по дългия път на изучаване на съвременните езици за програмиране и технологии за разработка на софтуер. Това е книга, която учи на фундаменталните принципи на програмирането, които не са се променили съществено през последните 15 години.

Не се притес­нявайте да прочетете тази книга, дори C# да не е езикът, с който искате да се занимавате. С който и друг език да продължите по-нататък, знанията, които ще ви дадем, ще ви останат трайно, защото тази книга ще ви научи **да мислите като програмисти**. Ще ви покажем и научим как да пишете програми, с които да решавате практически задачи по програми­ране, ще ви изградим умения да измисляте и реализирате алгоритми и да ползвате различни структури от данни.

Колкото и да ви се струва невероятно, **базовите принципи** на писане на компютърни програми не са се променили съществено през последните 15 години. Езиците за програмиране се променят, технологиите се променят, средствата за разработка се развиват, но принципите на програмирането си остават едни и същи. Когато човек се научи **да мисли алгоритмично**, когато се научи инстинктивно да разделя проблемите на последовател­ност от стъпки и да ги решава, когато се научи да подбира подходящи структури от данни и да пише **качествен програмен код**, тогава той става програмист. Когато придобиете тези умения, лесно можете да научите нови езици и различни технологии, като уеб програмиране, бази от данни, HTML5, XML, SQL, ASP.NET, XAML, Flash, Java EE и още стотици други.

Тази книга е именно за това да ви научи да мислите като програмисти, а езикът C# е само един инструмент, който може да се замени с всеки друг съвременен програмен език, например Java, C++, PHP или Python. **Това е книга за програмиране, а не книга за C#!**

## За кого е предназначена тази книга?

Тази книга е най-подходяща за **начинаещи**. Тя е предназначена за всички, които не са се занимавали до момента сериозно с програмиране и имат желание да започнат. Тази книга стартира от нулата и ви запознава стъпка по стъпка с основите на програмирането. Тя няма да ви научи на всичко, което ви трябва, за да станете **софтуерен инженер** и да работите в софтуерна фирма, но ще ви даде основи, върху които да градите техноло­гични знания и умения, а с тях вече ще можете да превърнете програми­рането в своя професия.

Ако никога не сте писали компютърни програми, не се притеснявайте. Винаги има първи път. В тази книга **ще ви научим на програмиране от нулата**. Не очакваме да знаете и можете нещо предварително. Доста­тъчно е да имате компютърна грамотност и желание да се занимавате с програ­миране. Останалото ще го прочетете от тази книга.

Ако вече можете да пишете прости програмки или сте учили програмиране в училище, или в университета, или сте писали програмен код с приятели, не си мислете, че знаете всичко! **Прочетете тази книга** и ще се убедите колко много неща сте пропуснали. Книгата е за начинаещи, но ви дава концепции, които дори някои програмисти с богат опит не владеят. В софтуер­ните фирми са се навъдили възмутително много самодейци, които въп­реки че програмират на заплата от години, не владеят основите на прог­рамирането и не знаят какво е **хеш-таблица**, как работи **полиморфиз­мът** и как се работи с **побитови операции.** Не бъдете като тях! Научете първо основите на програмирането, а след това технологиите. В противен случай рискувате да останете осакатени като програмисти за много дълго време (а може би и за цял живот).

Ако пък имате **опит с програмирането**, за да прецените дали тази книга е за вас, я разгледайте подробно и вижте дали са ви познати всички теми, които сме разгледали. Обърнете по-голямо внимание на главите "[Структури от данни – съпоставка и препоръки](#_Глава_19._Структури)", "[Принципи на обектно-ориен­тираното програми­ране](#_Глава_20._Принципи)", "[Как да решаваме задачи по програ­миране?](#_Глава_23._Как)" и "[Качествен програмен код](#_Глава_21._Качествен)". Много е вероятно, дори ако имате няколко години опит, да не владеете добре работата със **структури от данни**, да не умеете да оценявате **сложност на алгоритъм**, да не вла­деете в дълбочина концепциите на обектно-ориентираното програмиране (включи­телно UML и design patterns) и да не познавате добрите практики за писане **на качествен програмен код**. Това са много важни теми, които не се срещат във всяка книга за програмиране, така че не ги пропускайте!

### Не са необходими начални познания

В тази книга не очакваме от читателите да имат предварителни знания по програмиране. Не е необходимо да сте учили информационни технологии или компютърни науки, за да четете и разбирате учебния материал. Книгата **започва от нулата** и постепенно ви въвлича в програми­рането. Всички технически понятия, които ще срещнете, са обяснени преди това и не е нужно да ги знаете от други източници. Ако не знаете какво е компила­тор, дебъгер, среда за разработка, променлива, масив, цикъл, конзола, символен низ, структура от данни, алгоритъм, сложност на алгоритъм, клас или обект, не се плашете. От тази книга ще научите всички тези понятия и много други и постепенно ще свикнете да ги ползвате непрестанно в ежедневната си работа. Просто четете книгата последователно и **правете упражненията**.

Ако все пак **имате предварителни познания** по компютърни науки и информационни технологии, при всички положения те ще са ви от полза. Ако следвате университетска специалност свързана с компютър­ните техно­логии или в училище учите информационни технологии, това само ще ви помогне, но не е задължително. Ако учите туризъм или право, или друга специалност, която няма много общо с компютърните технологии, също можете да станете добър програмист, стига да имате желание.

Би било полезно да имате **начална компютърна грамотност**, тъй като няма да обясняваме какво е файл, какво е твърд диск, какво е мрежова карта, как се движи мишката и как се пише на клавиатурата. Очакваме да знаете как да си служите с компютъра и как да ползвате Интернет.

Препоръчва се читателите да имат **някакви знания по английски език**, поне начални. Всичката документация, която ще ползвате ежедневно, и почти всички сайтове за програмиране, които ще четете постоянно, са на английски език. В професията на програмиста английският е абсолютно задъл­жителен. Колкото по-рано го научите, толкова по-добре.

|  |  |
| --- | --- |
|  | Не си правете илюзии, че можете да станете програмисти, без да научите поне малко английски език! Това е просто наивно очакване. Ако не знаете английски, преминете някакъв курс и след това започнете да четете технически текстове и си вадете непознатите думи и ги заучавайте. Ще се уверите, че техническият английски се учи лесно и не отнема много време. |

## Какво обхваща тази книга?

Настоящата книга обхваща **основите на програмирането**. Тя ще ви научи как да дефинирате и използвате променливи, как да работите с прими­тивни структури от данни (като например числа), как да организирате логически конструкции, условни конструкции и цикли, как да печатате на конзолата, как да ползвате масиви, как да работите с бройни системи, как да дефинирате и използвате методи и да създавате и използвате обекти. Наред с началните познания по програмиране книгата ще ви помогне да възприемете и малко **по-сложни концепции** като обра­ботка на символни низове, работа с изключения, използване на сложни структури от данни (като дървета и хеш-таблици), работа с текстови файлове, дефиниране на собствени класове и работа с LINQ заявки. Ще бъдат застъпени в дълбочина кон­цепциите на **обектно-ориентираното програмиране** като утвърден подход в съвре­менната разработка на софтуер. Накрая ще се сблъскате с практи­ките за писане на висококачест­вени програми и с решаването на реални проб­леми от програмирането. Книгата излага цялостна методо­логия за **реша­ване на задачи по програ­миране** и въобще на алгоритмични проб­леми и показва как се прилага тя на практика с няколко примерни теми от изпити по програмиране. Това е нещо, което няма да срещнете в никоя друга книга за програмиране.

## На какво няма да ви научи тази книга?

Тази книга **няма да ви даде професията "софтуерен инженер"**! Тази книга няма да ви научи да ползвате цялата .NET платформа, да работите с бази от данни, да правите динамични уеб сайтове, да боравите с прозоречен графичен потребителски интерфейс и да разработвате уеб приложения. Няма да се научите да разработвате сериозни софту­ерни приложения и системи като например Skype, Firefox, MS Word или социални мрежи като Facebook и търговски портали като Amazon.com. За такива проекти са нужни много, **много години работа и опит** и познанията от тази книга са само едно прекрасно начало.

От книгата **няма** да се научите на софтуерно инженерство и работа в екип и няма да можете да се подгот­вите за работа по реални проекти в софту­ерна фирма. За да се научите на всичко това ще ви трябват **още няколко книги** и допълнителни обучения, но не съжалявайте за времето, което ще отделите на тази книга. Правите правилен избор, като започвате от осно­вите на програмирането вместо директно от уеб и мобилни приложения и бази данни. Това ви дава шанс **да станете добър програмист**, който разбира програмирането и технологиите в дълбочина. След като усвоите основите на програмирането, ще ви е много по-лесно да четете и учите за бази данни и уеб приложения и ще разбирате това, което четете, много по-лесно и в много по-голяма дълбочина, отколкото ако се захванете да учите директно SQL, ASP.NET, AJAX, WPF или мобилна разработка.

Някои ваши колеги започват да програмират директно от уеб приложения и бази от данни, без да знаят какво е масив, какво е списък и какво е хеш-таблица. Не им завиждайте! Те са тръгнали по трудния път, отзад напред. Ще се научат да правят **нискокачествени уеб сайтове** с PHP и MySQL, но ще им е безкрайно трудно да станат истински професиона­листи. И вие ще научите уеб технологиите и базите данни, но преди да се захванете с тях, **първо се научете да програмирате**. Това е много по-важно. Да научите една или друга технология е много лесно, след като имате основата, след като можете да мислите алгоритмично и знаете как да подхождате към пробле­мите на програмирането.

|  |  |
| --- | --- |
|  | Да започнеш с програмирането от уеб приложения и бази данни е също толкова неправилно, колкото и да започнеш да учиш чужд език от някой класически роман вместо от буквар или учебник за начинаещи. Не е невъзможно, но като ти липсват основите, е много по-трудно. Възможно е след това с години да останеш без важни фундаментални знания и да ставаш за смях пред колегите си. |

## Как е представена информацията?

Въпреки големия брой автори, съавтори и редактори, сме се постарали стилът на текста в книгата да бъде **изключително достъпен**. Съдържанието е представено в добре структуриран вид, разделено с множество заглавия и подзаглавия, което позволява лесното му възприемане, както и бързото търсене на инфор­мация в текста.

Настоящата книга е написана **от програмисти за програмисти**. Авто­рите са действащи софтуерни разработчици, колеги с реален опит както в разра­ботването на софтуер, така и в обучението по програмиране. Благо­даре­ние на това качеството на изложението е на мно­го добро ниво.

Всички автори ясно съзнават, че **примерният сорс код** е едно от най-важните неща в една книга за програ­ми­ране. Именно поради тази причи­на текстът е съпроводен с много, много примери, илюстрации и картинки.

Няма как, когато **всяка глава е писана от различен автор**, да няма разминавания между стиловете на изказ и качеството на отделните глави. Някои автори вложиха много старание (месеци наред) и много усилия, за да станат перфектни техните глави. Други не вложиха чак толкова усилия и затова някои глави не са така хубави и изчерпателни като другите. Не на последно място опитът на авторите е различен: някои програмират професионално от 2-3 години, докато други – от 15 години насам. Няма как това да не се отрази на качеството, но ви уверяваме, че всяка глава е минала редакция и отговаря поне минимално на високите изисквания на водещите автори на книгата – Светлин Наков и Веселин Колев.

## Какво е C#?

Вече обяснихме, че тази книга **не е за езика C#**, а за **програмирането** **като** **концепция** и неговите основни принципи. Ние използваме езика C# и платформата Microsoft .NET Framework само като средство за писане на програмен код и не наблягаме върху спецификите на езика. Настоящата книга може да бъде намерена и във варианти за други езици като **Java** и C++, но разликите не са съществени.

Все пак, нека разкажем с няколко думи какво е **C#** (чете се "**си шарп**").

|  |  |
| --- | --- |
|  | C# e съвременен език за програмиране и разработка на софтуерни приложения. |

Ако думичките "C#" и ".NET Framework" ви звучат непознато, в [следва­щата глава](#_Глава_1._Въведение) ще научите подробно за тях и за връзката между тях. Сега нека все пак обясним съвсем накратко какво е C#, какво е .NET, .NET Framework, CLR и останалите свързани с езика C# технологии.

### Езикът C#

C# е **съвременен обектно-ориентиран език за програмиране** с общо пред­назначение, създаден и развиван от Microsoft като част от **.NET платформата**. На езика C# и върху .NET платформата се разработва изключително разнообразен софтуер: офис приложения, уеб приложения и уеб сайтове, настолни приложения, мултимедийни приложе­ния, при­ложения за мобилни телефони и таблети, игри и много други.

**C# е език от високо ниво**, който прилича на Java и C++ и донякъде на езици като Delphi, VB.NET и C. Всички C# програми са обектно-ориенти­рани. Те представляват съвкупност от дефиниции на класове, които съдържат в себе си методи, а в методите е разположена програмната логика – инструкциите, които компютърът изпълнява. Повече детайли за това какво е клас, какво е метод и какво представляват C# програмите ще научите в [следващата глава](#_Глава_1._Въведение).

В днешно време **C# е един от най-популярните езици за програмиране**. На него пишат милиони разработчици по цял свят. Тъй като C# е разработен от Майкро­софт като част от тяхната съвременна платформа за разработка и изпъл­нение на приложения .NET Framework, езикът е силно разпростра­нен сред Microsoft-ориентираните фирми, организации и индивидуални разработчици.

Езикът C# и платформата .NET Framework се поддържат и контро­лират от Microsoft, но постепенно се отварят и за външния свят чрез миграция към **отворен код** с усилията на .NET фондацията ([www.dotnetfoundation.org](http://www.dotnetfoundation.org)) стартиала през 2014 г. Поради години наред затворено развитие на езика C# и .NET платформата останали големи световни софтуерни корпорации като Google, Apple, IBM, Oracle и SAP базират своите решения на Java или други платформи и не използват C# като основен език за разработка на своите продукти.

Езикът C# се разпро­странява заедно със специална среда, върху която се изпълнява, наре­чена **Common Language Runtime (CLR)**. Тази среда е част от **платформата .NET Framework**, която включва CLR, пакет от стандартни библиотеки, предоставящи базова функционалност, компила­тори, дебъгери и други средства за разработка. Благодарение на нея CLR програмите са преносими и след като веднъж бъдат написани, могат да работят почти без промени върху различни хардуерни платформи и операционни системи. Най-често C# програмите се изпълняват върху MS Windows, но .NET Framework и CLR се поддържа и за мобилни телефони и други преносими устройства, базирани на Windows Mobile. Под Linux, FreeBSD, MacOS X и други операционни системи C# програмите могат да се изпъл­няват върху свободната .NET Framework имплементация **Mono**, която обаче не се поддържа официално от Microsoft. Тенденциите (към 2015 г.) са Microsoft постепенно да започне да поддръжа C# и .NET под Linux.

### Платформата Microsoft .NET Framework

Езикът C# не се разпространява самостоятелно, а е част от платформата Microsoft .NET Framework (чете се "**майкрософт** **дот нет фреймуърк**"). .NET Framework най-общо представлява среда за разработка и изпъл­нение на програми, написани на езика C# или друг език, съвместим с .NET (като VB.NET, Managed C++, J# или F#). Тя се състои от .NET езици за програмиране (C#, VB.NET и други), **среда за изпълне­ние на управ­ляван код (CLR)**, която изпълнява контролирано C# програ­мите, и от съвкупност от стандартни библиотеки и инструменти за разра­ботка, като например **компилаторът** csc, който превръща C# програ­мите в разбираем за CLR междинен код (наречен MSIL) и библиотеката ADO.NET, която осигурява достъп до бази от данни (например MS SQL Server или MySQL). .NET Framework е част от всяка съвременна Windows дистрибуция и може да се срещне в различни свои версии. Последна версия може да се изтегли и инсталира от сайта на Microsoft.

## Защо C#?

Има много причини да изберем езика C# за нашата книга. Той е **съвре­менен език за програмиране**, широкоразпространен, използван от мили­они програмисти по целия свят. Същевременно C# е **изключително прост и лесен за научаване език** (за разлика от C и C++). Нормално е да започнем от език, който е подходящ за начинаещи и се ползва много в практиката. Именно такъв език избрахме – лесен и много популярен, език, който се ползва широко в индустрията от много големи и сериозни фирми.

### C# или Java или друг език?

Въпреки че по този въпрос **може много да се спори**, се счита, че най-сериозният съперник на C# е Java. Няма да правим сравнение между Java и C#, тъй като C# е безспорно по-добре развития, по-мощният и **по-добре измисленият от инженерна гледна точка език**, но трябва да обърнем внимание, че за целите на настоящата книга всеки съвременен език за програмиране ще ни свърши работа. Ние избрахме C#, защото е по-лесен за изучаване и се разпространява с изключително удобни безплатни среди за разработка (например Visual Studio Community Edition). Който има предпочи­тания към Java може да ползва [**Java варианта на настоящата книга**](http://www.introprogramming.info/intro-java-book/), достъпен от нейния сайт: [www.introprogramming.info](http://www.introprogramming.info).

### Защо не PHP?

По отношение на популярност освен C# и Java **много широко използван** е езикът PHP. Той е подходящ за разработка на малки уеб сайтове и уеб приложения, но създава сериозни трудности при реализацията на големи и сложни софтуерни системи. В софтуерната индустрия PHP се ползва предимно за малки проекти, тъй като предразполага към писане на лош, **неоргани­зиран и труден за поддръжка код**, поради което е неудобен за по-сериозни проекти. По този въпрос може много да се спори, но се счита, че поради **остарелите** си концепции и подходи, върху които е построен, и поради редица **еволюционни причини** PHP е език, който пред­раз­полага към некачествено програмиране и изграж­дане на труден за поддръжка софтуер. По тази причина ви препоръчваме да започнете да програмирате не от PHP, а от C#, Java, Python или друг съвременен език.

**PHP** **има своето приложение** в света на програмирането (например да си направим блог с WordPress, малък сайт с Joomla или Drupal или дискусионен форум с PhpBB), но цялата PHP платформа не е така зряла и добре организирана като .NET и Java ЕЕ. Когато става дума за не-уеб базирани приложения, мобилна разработка или големи индустриални проекти, PHP изобщо не е сред възможностите за избор. За да се ползва коректно PHP и да се разработват профе­сионални проекти с **високо качество** е необходим много, много опит. Обикновено PHP разработчиците учат от самоучители, статии и книги с ниско качество и заучават вредни прак­тики и навици, които след това е много трудно да се изчистят. Затова не учете PHP като ваш пръв език за разработка. Започнете със C# или Java.

На базата на огромния опит на авторския колектив можем да ви **препоръчаме** да започнете да учите програмиране с езика **C#**, като пропуснете езици като C, C++ и PHP до момента, в който не ви се наложи да ги ползвате.

### Защо не C или C++?

Въпреки че отново много може да се спори, езиците **C и C++** се считат за доста примитивни, остарели и отми­ращи. Те все пак имат своето приложение и са подходящи за **програмиране на ниско ниво** (например за специализирани хардуерни устройства), но не ви съветваме да се занимавате с тях докато се учите да програмирате.

**Чисто C** се използва за **ситемно програмиране** – да програмирате, ако трябва да пишете операционна система, драйвер за хардуерно устройство или да програ­мирате про­мишлен контролер (embedded device), поради липса на алтернатива и поради нуждата да се управлява много внимателно хардуерът. **Езикът C е** **морално остарял** и в никакъв случай не ви съветваме да започвате да учите програмиране с него. Производителността на програмиста при разработка на чисто C е в пъти по-ниска отколкото при съвременните езици за програмиране с общо предназначение като C# и Java. Вариант на езика C се използва при Apple / iPhone разработчиците, но не защото е хубав език, а защото няма свястна алтернатива. Повечето Apple-ориенти­рани разработчици не харесват Objective-C, но почти нямат избор да пишат на нещо друго. За щастие Apple обявиха постепенна миграция към новия им език **Swift** през 2014 г.

**C++** е добър, когато трябва да програмирате определени приложения, които изискват много **близка работа с хардуера** или имат специални изисквания за **бързо­действие** (например разработка на 3D игри). За всички останали задачи (например разработка на уеб приложения или бизнес софтуер) C++ е изключително **неподходящ**. Не ви съветваме да се захващате с него, ако сега стартирате с програмирането. Причината все още да се учи C++ в някои училища и университети е наследствена, тъй като тези институции са доста консервативни. Например към 2014 г. международната олимпи­ада по информатика за ученици (IOI) продължава да промотира C++ като единствения език, позволен за използване по състезанията по програмиране, въпреки че **в индустрията C++ не се използвa масово**. Ако не вярвате, разгледайте някой сайт с обяви и пребройте колко процента от обявите за работа изискват C++.

Езикът C++ изгуби своята популярност най-вече поради **невъзможността на него да се разработва бързо** качествен софтуер. За да пишете кадърно на C++, трябва да сте много печен и опитен програмист, докато за C# и Java не е чак толкова задължително. **Ученето на C++ отнема в пъти повече време** и много малко програмисти го владеят наистина добре. Производителността на C++ програмистите е в пъти по-ниска от тази на C# и затова C++ все повече губи позиции. Поради всички тези причини този език постепенно си отива и затова не ви съветваме да го учите.

### Предимствата на C#

C# е **обектно-ориентиран** език за програмиране. Такива са всички съвре­менни езици, на които се разработват сериозни софтуерни системи (например Java и C++). За предимствата на обектно-ориентираното програмиране (ООП) ще стане дума подробно на много места в книгата, но за момента може да си представяте обектно-ориентираните езици като езици, които позволяват да работите с обекти от реалния свят (например студент, училище, учебник, книга и други). Обектите имат характеристики (например име, цвят и т.н.) и могат да извършват действия (например да се движат, да говорят и т.н.).

Започвайки с програмирането от езика C# и платформата .NET Framework, вие поемате по един **много перспективен път**. Ако отворите някой сайт с обяви за работа за програмисти, ще се убедите, че търсенето на C# и .NET специалисти е огромно и е близко до обема на търсенето на Java програ­мисти. Същевременно търсенето на специалисти по PHP, C++ и всички останали технологии е много по-малко, отколкото търсенето на C# и Java инженери.

За добрия програмист езикът, на който пише, няма съществено значение, защото той умее да програмира. Каквито и езици и технологии да му трябват, той **бързо ги овладява**. Нашата цел е не да ви научим на C#, а да ви научим на програмиране! След като овладеете основите на програ­мирането и се научите да мислите алгоритмично, можете да научите и други езици и ще се убедите колко много приличат те на C#. Програмира­нето се гради на принципи, които много бавно се променят с годините и тази книга ви учи точно на тези принципи.

## Примерите са върху C# 4.0 и Visual Studio 2010

Всички примери в книгата се отнасят за версия 4.0 на езика C# и плат­формата .NET Framework, която към момента на първото публикуване на книгата (2011 г.) е последната. Всички примери за използване на средата за разработка **Visual Studio** се отнасят за версия 2010 на продукта, която е последна към момента на публи­куване на книгата. Разбира се, препоръчваме ви да ползвате последната версия на Visual Studio, която е налична от Microsoft. Може да проверите тук: [www.visualstudio.com](https://www.visualstudio.com).

Средата за разработка Microsoft Visual Studio 2010 **има безплатна версия**, подходяща за начинаещи C# програмисти, наречена **Microsoft Visual Studio Community Edition**, но разликата между Community Edition и пълната версия на Visual Studio (която е комерсиален софтуерен продукт) е предимно в лиценза и не касае темите от настоящата книга.

Коя версия на C# и Visual Studio ще използвате докато се учите да програ­мирате **не е от съществено значение**. Важното е да научите **принципите на програмирането и алгоритмичното мислене**! Езикът C#, платфор­мата .NET Framework и средата Visual Studio са само едни инструменти и можете да ги замените с други по всяко време.

## Как да четем тази книга?

Четенето на тази книга трябва да бъде съпроводено с много, **много практика**. Няма да се научите да програмирате, ако не практикувате! Все едно да се научите да плувате от книга, без да пробвате. Няма начин! Колкото повече работите върху задачите след всяка глава, толкова повече ще научите от книгата.

Всичко, което прочетете тук, трябва **да изпробвате сами** на компютъра. Иначе няма да научите нищо. Например, когато прочетете за Visual Studio и как да си направите първата проста програмка, трябва **непременно** да си изтеглите и инсталирате Microsoft Visual Studio и да пробвате да си направите някаква програмка. Иначе няма да се научите! **На теория винаги е по-лесно**, но **програмирането е практика**. Запомнете това и решавайте задачите за упражнения от книгата. Те са внимателно подбрани – хем не са много трудни, за да не ви откажат, хем не са много лесни, за да ви мотивират да приемете решава­нето им като предизвика­телство.

Ако имате трудности със задачите или с учебния материал, потърсете помощ във форума на **Софтуерния университет** (СофтУни), където хиляди студенти учат по настоящата книга и ще ви помогнат за всяка задача от нея: <http://softuni.bg/forum/>.

|  |  |
| --- | --- |
|  | Четенето на тази книга без практика е безсмислено! Трябва да отделите за писане на програми няколко пъти повече време, отколкото отделяте да четете текста. |

Всеки е учил математика в училище и знае, че за да се научи да решава задачи по математика, му трябва **много практика**. Колкото и да гледа и да слуша учителя, без да седне да решава задачи, никой не може да се научи. Така е и с програмирането. Трябва ви много практика. Трябва да пишете много, **да решавате задачи**, да експериментирате, да се мъчите и да се борите с проблемите, да грешите и да се поправяте, да пробвате и да не става, да пробвате пак по нов начин и да изживеете моментите, в които се получава. Трябва ви **много, много практика**. Само така ще напреднете.

### Не пропускайте упражненията!

На края на всяка глава има сериозен списък със **задачи за упражнения**. Не ги пропускайте! Без упражненията нищо няма да научите. След като прочетете дадена глава, трябва да седнете на компютъра и **да пробвате примерите**, които сте видели в книгата. След това трябва да се хванете и **да решите всички задачи**. Ако не можете да решите всички задачи, трябва поне да се помъчите да го направите. Ако нямате време, трябва да решите поне първите няколко задачи от всяка глава. Не преминавайте напред, без **да решавате задачите** след всяка глава! Просто няма смисъл. Задачите са малки реални ситуации, в които прилагате прочетеното. В практиката, един ден, когато станете програмисти, ще решавате всеки ден подобни задачи, но по-големи и по-сложни.

|  |  |
| --- | --- |
|  | Непременно решавайте задачите за упражнения след всяка глава от книгата! Иначе рискувате нищо да не научите и просто да си загубите времето. |

### Колко време ще ни трябва за тази книга?

Усвояването на основите на програмирането е **много сериозна** задача и отнема много време. Дори и силно да ви се отдава, **няма начин да се научите да програмирате на добро ниво за седмица или две**. За научава­нето на всяко човешко умение е необходимо да прочетете или да видите, или да ви покажат как се прави и след това да пробвате сами. При прог­рамирането е същото – трябва или да прочетете, или да гледате, или да слушате как се прави, след това да пробвате сами и да успеете или да не успеете и да пробвате пак и накрая да усетите, че сте го научили. **Ученето става стъпка по стъпка**, последователно, на малки порции, с много усърдие и постоянство – месеци, дори години наред, всеки ден.

Ако искате да прочетете, разберете, научите и усвоите цялостно и в дъл­бочина целия учебния материал от тази книга, ще трябва да инвес­тиратепоне **2-3 месеца целодневно** или поне **5-6 месеца, ако четете и се упраж­нявате по малко всеки ден**. Това е минималното време, за което можете да усвоите в дълбочина основите на програмирането.

Основният учебен материал в книгата е изложен в около 1100 страници, за които ще ви трябва **поне месец** (по цял ден), само за да го прочетете внима­телно и да изпробвате примерните програми. Разбира се, трябва да отделите достатъчно внимание и на упражненията, защото без тях почти нищо няма да научите.

Упражненията съдър­жат около **350 задачи** с различна трудност. За някои от тях ще ви трябват по няколко минути, докато за други ще ви трябват по няколко часа (ако въобще успеете да ги решите без чужда помощ). Това означава, че ще ви трябва **месец-два по цял ден да се упражнявате** или да го правите по малко в продължение на няколко месеца.

Ако не разполагате с толкова време, замислете се дали наистина искате **да се занимавате с програмиране**. Това е много сериозно начинание, в което трябва да вложите наистина много усилия. Ако наистина искате да се научите да програмирате на добро ниво, планувайте си достатъчно време и следвайте книгата.

## Защо фокусът е върху структурите от данни и алгоритмите?

Настоящата книга, наред с основните познания по програмиране, ви учи и на правилно алгоритмично мислене и работа с основните структури от данни в програмирането. **Структурите от данни и алгоритмите са най-важните фундаментални знания на един програмист**! Ако ги овладеете добре, след това няма да имате никакви проблеми да овладеете която и да е софтуерна технология, библиотека, framework или API. Именно на това разчитат и най-сериозните софтуерни фирми в света, когато наемат служители. Потвърждение са интервютата в големите фирми като Google и Microsoft, които изключително много държат на правилното алгоритмично мислене и познаването на всички базови структури от данни и алгоритми.

### Интервютата за работа в Google

На интервютата за работа като софтуерен инженер в Google в Цюрих 100% от въпросите са съсредоточени върху **структури от данни, алго­ритми и алгоритмично мислене**. На такова интервю могат да ви накарат да реализирате на бяла дъска свързан списък (вж. главата "[Линейни струк­тури от данни](#_Глава_16._Линейни)") или да измис­лите алгоритъм за запълване на растерен многоъгълник (зададен като GIF изображение) с даден цвят (вж. метод на вълната в главата "[Дървета и графи](#_Глава_17._Дървета)". Изглежда Google ги интересува да наемат хора, които имат **алгоритмично мислене** и владеят основните структури от данни и базовите компютърни алгоритми. Всички технологии, които избраните кандидати ще използват след това в рабо­тата си, могат бързо да бъдат усвоени. Разбира се, не си мислете, че тази книга ще ви даде всички знания и умения, за да преминете успешно интервю за работа в Google. **Знанията от книгата са абсолютно необхо­дими**, но не са достатъчни. Те са само първите стъпки.

### Интервютата за работа в Microsoft

На интервютата за работа като софтуерен инженер в Microsoft в Дъблин голяма част от въпросите са съсредоточени върху **структури от данни, алго­ритми и алгоритмично мислене**. Например могат да ви накарат да обърнете на обратно всички думи в даден символен низ (вж. главата "[Символни низове](#_Глава_13._Символни)") или да реализирате топологично сортиране в неори­ентиран граф (вж. главата "[Дървета и графи](#_Глава_17._Дървета)"). За разлика от Google в Microsoft питат и за много **инженерни въпроси**, свързани със софтуерни архитектури, процеса на разработка на софтуер, паралелна обработка (multithreading), писане на сигурен код, работа с много големи обеми от данни и тестване на софтуера. Настоя­щата книга далеч не е достатъчна, за да кандидатствате в Microsoft, но със сигурност знанията от нея ще ви бъдат полезни за голяма част от въпросите.

### За технологията LINQ

В книгата е включена една тема за популярната технология LINQ (Language Integrated Query), която позволява изпълнение на различни заявки (като търсене, сортиране, сумиране и други групови операции) върху масиви, списъци и други обекти. Тя нарочно е разположена към края, след темите за структури от данни и сложност на алгоритми. Причината за това е, че добрият програмист **трябва да знае какво се случва**, когато сортира списък или търси по даден критерий в масив и колко операции отнемат тези действия. Ако се използва LINQ, не е оче­видно как работи дадена заявка и колко време отнема. **LINQ и функционалното изразяване с ламбда изрази са много мощна** и широко използвана технология, но тя трябва да бъде овладяна на по-късен етап, след като познавате добре основите на програмирането и основните **алгоритми и структури от данни**. В противен случай риску­вате да се научите да пишете неефективен код, без да си давате сметка как работи той и колко операции извършва.

## Наистина ли искате ли да станете програмист?

Ако искате да станете програмист, трябва да знаете, че **истинските програмисти** са сериозни, упорити, мислещи и търсещи хора, които се справят с всякакви задачи, и за тях е важно да могат бързо да овладяват всички необходими за работата им платформи, технологии, библиотеки, прог­рамни средства, езици за програмиране и инструменти за разработка и да усещат програмирането като част от себе си.

Добрите програмисти отделят изклю­чително много време **да развиват инженерното си мислене**, да учат ежедневно нови технологии, нови езици за програмиране, нови начини на работа, нови платформи и нови средства за разработка. Те умеят да **мислят логически**, да разсъждават върху проблемите и да измислят алгоритми за решаването им, да си представят решенията като последователност от стъпки, да моделират заобикалящия ги свят със средствата на технологиите, да реализират идеите си като **програми** или програмни компоненти, да тестват алгорит­мите и програмите си, да виждат проблемите, да предвиждат изключи­телните ситуации, които могат да възникнат и да ги обработват правилно, да се вслушват в съветите на по-опитните от тях, да съобразяват потреби­телския интерфейс на програмите си с потребителя и да съобразяват алго­ритмите си с възможностите на машините, върху които те се изпъл­няват, и със средата, в която работят и с която си взаимодействат.

Добрите програмисти непрекъснато четат книги, статии или блогове за прог­рамиране и се интересуват от новите технологии, **постоянно обога­тяват познанията си** и подобряват начина на работата си и качеството на написания от тях софтуер. Някои от тях се вманиачават до такава степен, че забравят да ядат или спят, когато имат да решат сериозен проблем или просто са вдъхновени от някоя нова технология или гледат някоя инте­ресна лекция или презентация. Ако вие имате склонност да се мотивирате до такава степен в едно нещо (например да играете денонощно компю­търни игри), можете **бързо да се научите да програмирате**, стига просто да се настроите, че най-интересното нещо на света за вас в този период от живота ви е програмирането.

Добрите програмисти имат един или няколко компютъра и Интернет и живеят в **постоянна връзка с технологиите**. Те посещават редовно сайтове и блогове, свързани с новите технологии, комуникират ежедневно със свои колеги, посещават технологични лекции, семинари и други събития, следят технологичните промени и тенденции, пробват новите технологии, дори и да нямат нужда от тях в момента, **експериментират** и проучват новите възможности и новите начини да направят даден софтуер или елемент от работата си, разглеждат нови библиотеки, учат нови езици, пробват нови технологични рамки (frameworks) и си играят с новите средства за разработка. Така те **развиват своите умения** и поддържат доброто си ниво на информираност, компетентност и професионализъм.

Истинските програмисти знаят, че никога не могат да овладеят профе­сията си до краен предел, тъй като тя се променя постоянно. Те живеят с твърдото убеждение, че **цял живот трябва да учат**, и това им харесва и им носи удовлетворение. Истинските програмисти са **любопитни** и търсещи хора, които искат **да знаят как работи** – от обикновения аналогов часовник, до GPS системата, Интернет технологиите, езиците за програ­миране, операционните системи, компилаторите, компютърната графика, игрите, хардуерът, изкуствения интелект и всичко останало, свързано с компютрите и технологиите. Колкото повече научават, толкова повече са **жадни за още знания и умения**. Животът им е свързан с технологиите и те се променят заедно с тях, наслаждавайки се на развитието на компютър­ните науки, технологиите и софтуерната индустрия.

Всичко, което ви разказваме за **истинските програмисти**, го знаем от личен опит и сме убедени, че програмист е професия, която иска да й се посветиш и **да й се отдадеш напълно**, за да бъдеш наистина добър специалист. Истинският програмист трябва да е опитен, компетентен, информиран, мислещ, разсъждаващ, знаещ, можещ, умеещ да се справя в нестандартни ситуации. Всеки, който се зах­ване с програмиране "помежду другото" е обречен да бъде посред­ствен програмист. Програмирането изисква **пълно себеотдаване в продъл­жение на години**. Ако сте готови за всичко това, продължавайте да четете напред и си дайте сметка, че тези няколко месеца, които ще отделите на тази книга за програмиране са само едно малко начало, а след това години наред ще учите **докато превърнете програмирането в своя профе­сия**, а след това ежедневно ще учите по нещо и ще се състезавате с технологиите, за да поддържате нивото си, докато някой ден програми­рането ви даде доста­тъчно развитие на мисленето и уменията ви, за да се захванете с друга професия, защото са малко програ­мистите, които програмират до пенсия, но са наистина много успешните хора, стартирали кариерата си с програмиране.

Ако все още не сте се отказали да станете добър програмист и вече сте си изградили дълбоко в себе си разбиране, че **следващите месеци и години от живота ви ще бъдат ежедневно свързани с постоянен усърден труд** по овладяване на тайните на програмирането, разработката на софтуер, компютърните науки и софтуерните технологии, може да използвате една стара **техника за вътрешна мотивация** и уверено постигане на цели, която може да бъде намерена в някои книги и древни учения под една или друга форма: представяйте си постоянно, че сте **програмисти**, че сте успели да станете такива, че се занимавате ежедневно с програмиране и то е **вашата професия** и че можете да напишете целия софтуер на света (стига да имате достатъчно време), че умеете да решите всяка задача, която опитните програмисти могат да решат, и си мислете постоянно и непрекъснато за вашата цел. Казвайте на себе си, дори понякога на глас: "**аз искам да стана добър програмист** и трябва много да работя за това, трябва много да чета и много да уча, трябва **да решавам много задачи, всеки ден, постоянно и усърдно**". Сложете книгите за програмиране нав­сякъде около вас, дори залепете надпис "аз ще стана добър програ­мист" над леглото си, така че да го виждате всяка вечер, когато лягате и всяка сутрин, когато ставате. **Програмирайте всеки ден**, решавайте задачи, забавлявайте се, учете нови технологии, експериментирайте, пробвайте да напишете игра, да направите уеб сайт, да напишете компилатор, база данни и още стотици програми, за които ви хрумнат оригинални идеи. За да станете добри програмисти, програмирайте всеки ден и **всеки ден мислете за програмирането** и си представяйте бъдещия момент, в който вие сте отличен програмист. Можете, ако дълбоко вярвате, че можете. Всеки може, стига да вярва, че може и да следва целите си постоянно, без да се отказва. Никой не може да ви мотивира по-добре от вас самите. Всичко зависи от вас и тази книга е първата ви стъпка.

## За НАРС, Telerik Academy и СофтУни

Водещият автор на книгата Светлин Наков ([www.nakov.com](http://www.nakov.com)) се занимава с обучение на софтуерни инженери от 2000 г. насам и е създател на няколко **учебни центъра за софтуерни инженери**, през които са преминали десетки хиляди млади хора: Национална академия по разработка на софтуер (НАРС), инициативата Telerik Academy и Софтуерния университет (СофтУни) – <http://softuni.bg>.

### Национална академия по разработка на софтуер (НАРС)

Първоначално книгата "Въведение в програмирането с Java" (предшест­веник на настоящата книга) възниква като проект на Светлин Наков за изграждане на **учебник по програмиране за начинаещи** за студентите от НАРС. Светлин Наков заедно с колеги изгражда първия по-сериозен учебен център за софтуерни инженери "НАРС" през 2005 г. През него преминават **над 600 студента**, които изучават безплатно програмиране, Java и .NET технологии и постъпват на работа в сериозни фирми от това време като SAP, Telerik, Johnson Controls, VMWare, Euro Games Technology (EGT), Musala Soft, Stemo, Rila Solutions, Sciant, Micro Focus, InterConsult Bulgaria (ICB), Acsior, Fadata, Seeburger Informatik и др. През 2009 г. Светлин Наков се оттегля и постепенно НАРС загубва позиции.

### Telerik Academy

През ноември 2009 г. Светлин Наков е поканен от световноизвестната софтуерна корпорация Телерик за да изгради учебен център за практическо обучение на софтуерни инженери. Така се ражда проектът "**Софтуерна академия на Телерик**" (Telerik Academy). Академията обучава безплатно **хиляди софтуерни инженери**, изгражда безплатно учебно съдържание за над 20 курса по програмиране и софтуерни технологии, споделя хиляди видео уроци в YouTube и дава професия и работа на хиляди младежи. Софтуерната академия организира подготовки за олимпиадите по информатика (НОИ) и по информа­ционни технологии (НОИТ) и занимания по програмиране за деца от 4-ти клас нагоре.

**Telerik Academy** (<http://telerikacademy.com>) започва с курс по C# и .NET технологии с 40 души през 2009 г. ръководен от Светлин Наков. Обученията постепенно се разширяват към стотици и дори **хиляди участници**. Изграждат се по-големи учебни зали и Светлин Наков заедно с екипа си разработва систематичен подход за **масирани обучения**, който позволява през следващите няколко години да се приемат под **над 1000 души годишно**. От тях за около година интензивно обучение завършват около 15% отлично добре подготвени програмисти и инженери по качеството на софтуера, от които най-силните започват работа в Телерик.

Освен основната си програма по инициатива на Светлин Наков в рамките на проекта "Телерик академия" се организират безплатни технологични обучения за ученици ([**Училищна софтуерна академия**](http://academy.telerik.com/school-academy/)), академия по алгоритми за състезателно програмиране ([**Алго академия**](http://academy.telerik.com/algoacademy/)) и [**Академия по програмиране деца**](http://telerik-kids.com) за ученици 4-7 клас.

Софтуерната академия се доказва като най-значимата за времето си образователна инициатива в ИТ сектора, печели **десетки награди** и изгражда огромна популярност.

През 2013 г. Светлин Наков се оттегля от Telerik, за да създаде Софтуерния университет ([СофтУни](http://softuni.bg)).

### Софтуерен университет (СофтУни)

Софтуерният университет ([СофтУни](http://softuni.bg/)) е **най-мащабният учебен център за софтуерни инженери в България**. През него преминават десетки хиляди студенти всяка година. [СофтУни](http://softuni.bg/) отваря врати през 2014 г. като продължение на усилията на Светлин Наков масирано да изгражда **качествени софтуерни специалисти** чрез истинско образование, което комбинира фундаментални знания със съвременни софтуерни технологии и много практика.

Софтуерният университет дава качествено образование, професия, работа и бакалавърска диплома за програмисти, софтуерни инженери и ИТ специалисти. [СофтУни](http://softuni.bg) изгражда изключително успешно трайна **връзка между образование и индустрия** като си сътрудничи със стотици софтуерни фирми и осигурява работа и стажове на своите студенти и предоставя качествени специалисти за софтуерната индустрия.

Обученията в Софтуерния университет ([СофтУни](http://softuni.bg)) обхващат **най-търсените умения в обявите за работа** от ИТ сектора. Освен практическо усвояване на най-търсените езици и технологии за разработка на софтуер, студентите изграждат алгоритмично мислене, умения за учене и за решаване на проблеми. В [СофтУни](http://softuni.bg) се усвояват **фундаментални основи на компютърните науки** и софтуерното инженерство. Чрез гъвкава програма се изучават съвременни езици и платформи за разработка, обектно-ориентирано и функционално програмиране, бази данни, уеб услуги, front-end технологии, сървърни технологии, уеб разработка и мобилни приложения. Набляга се на **практическите умения** и екипната работа чрез много упражнения на живо, много домашни, практически екипни проекти и практически изпити.

Софтуерният университет стартира нови **безплатни курсове по програмиране за начинаещи** почти всеки месец. Можете да се запишете от неговия сайт: <http://softuni.bg>.

Информация за всички предстоящи и минали **курсове и обучения по разработка на софтуер** и съвременни софтуерни технологии можете да получите и от личния сайт на Светлин Наков: [www.nakov.com](http://www.nakov.com).

## Поглед към съдържанието на книгата

Нека сега разгледаме накратко какво ни предстои в следващите глави на книгата. Ще разка­жем по няколко изречения за всяка от тях, за да знаете какво ви очаква да научите.

### Глава 1. Въведение в програмирането

В главата "[Въведение в програмирането](#_Глава_1._Въведение)" ще разгледаме основните термини от програмирането и **ще напишем първата си прог­рама**. Ще се запознаем с това какво е програмиране и каква е връзката му с компютрите и програмните езици. Накратко ще разгледаме основните етапи при писането на софтуер. Ще направим **въведение в езика C#** и ще се запознаем с .NET Framework и технологиите, свързани с него. Ще разгле­даме какви помощни средства са ни необходими, за да можем да програмираме на C#. Ще използваме C#, за да **напишем първата си програма**, ще я **компилираме** и **изпълним** както от командния ред, така и от среда за разработка Microsoft Visual Studio. Ще се запознаем още и с MSDN Library – документацията на .NET Framework, която позволява по-нататъшно изследване на възможностите на езика.

Автор на главата е Павел Дончев, а редактори са Теодор Божиков и Светлин Наков. Съдържанието на главата е донякъде базирано на рабо­тата на Лъчезар Цеков от книгата "Въведение в програмирането с Java".

### Глава 2. Примитивни типове и променливи

В главата "[Примитивни типове и променливи](#_Глава_2._Примитивни)" ще разгледаме **прими­тивните типове** и променливи в C# – какво представляват и как се работи с тях. Първо ще се спрем на **типовете данни** – целочислени типове, реални типове с плаваща запетая, булев тип, символен тип, низов тип и обектен тип. Ще продължим с това какво е **променлива**, какви са нейните характеристики, как се декларира, как се присвоява стойност и какво е инициализация на променлива. Ще се запознаем и с типовете данни в C# – стойностни и референтни. Накрая ще се спрем на **литералите**, ще разберем какво представляват и какви литерали има.

Автори на главата са Веселин Георгиев и Светлин Наков, а редактор е Николай Василев. Съдържанието на цялата глава е базирано на работата на Христо Тодоров и Светлин Наков от книгата "Въведение в програми­рането с Java".

### Глава 3. Оператори и изрази

В главата "[Оператори, изрази и конструкции за управление](#_Глава_3._Оператори)" ще се запознаем с **операторите** и **действията**, които те извършват върху различните типове данни. Ще разясним приоритетите на операторите и ще се запознаем с групите оператори според броя на аргументите, които приемат и действията, които извършват. След това ще разгледаме **преоб­разуването на типове**, защо е нужно и как да работим с него. Накрая ще опишем и покажем какво представляват изразите в C# и как се използват.

Автори на главата са Дилян Димитров и Светлин Наков, а редактор е Марин Георгиев. Съдържанието на цялата глава е базирано на работата на Лъчезар Божков от книгата "Въведение в програмирането с Java".

### Глава 4. Вход и изход от конзолата

В главата "[Вход и изход от конзолата](#_Глава_4._Вход)" ще се запознаем с **конзолата** като средство за **въвеждане и извеждане на данни**. Ще обясним какво пред­ставлява тя, кога и как се използва, какви са принципите на пове­чето програмни езици за достъп до конзолата. Ще се запознаем с някои от възможностите на C# за взаимодействие с потребителя. Ще разгледаме основните потоци за входно-изходни операции Console.In, Console.Out и Console.Error, класа Console и използването на **форма­тиращи низове** за отпечатване на данни в различни формати. Ще разгле­даме как можем да преобразуваме текст в число (**парсване**), тъй като това е начинът да въвеждаме числа в C#.

Автор на главата е Илиян Мурданлиев, а редактор е Светлин Наков. Съдържанието на цялата глава е до голяма степен базирано на работата на Борис Вълков от книгата "Въведение в програмирането с Java".

### Глава 5. Условни конструкции

В главата "[Условни конструкции](#_Глава_5._Условни)" ще разгледаме **условните конструкции** в C#, чрез които можем да изпълняваме различни действия в зависимост от някакво условие. Ще обясним синтаксиса на **условните оператори** – if и if-else – с подходящи примери и ще разясним практичес­кото приложение на опера­тора за избор switch. Ще обърнем внимание на добрите практики, които е нужно да бъдат следвани с цел постигане на по-добър стил на програмиране при изпол­зването на вложени и други видове условни конструкции.

Автор на главата е Светлин Наков, а редактор е Марин Георгиев. Съдър­жанието на цялата глава е базирано на работата на Марин Георгиев от книгата "Въведение в програмирането с Java".

### Глава 6. Цикли

В главата "[Цикли](#_Глава_6._Цикли_1)" ще разгледаме **конструкциите за цикли**, с които можем да изпълняваме даден фрагмент програмен код многократно. Ще разгле­даме как се реализират повторения с условие (while и do-while **цикли**) и как се работи с for**-цикли**. Ще дадем примери за различните възмож­ности за дефиниране на цикъл, за начина им на конструиране и за някои от основните им приложения. Накрая ще покажем как можем да изпол­зваме няколко цикъла един в друг (**вложени цикли**).

Автор на главата е Станислав Златинов, а редактор е Светлин Наков. Съдържанието на цялата глава е базирано на работата на Румяна Топалска от книгата "Въведение в програмирането с Java".

### Глава 7. Масиви

В главата "[Масиви](#_Глава_7._Масиви)" ще се запознаем с масивите като средства за обра­ботка на поредица от еднакви по тип елементи. Ще обясним какво представляват **масивите**, как можем да **декларираме**, **съз­даваме** и **инициа­лизираме** масиви и как можем да осъществяваме достъп до техните еле­менти. Ще обърнем внимание на **едномерните** и **много­мерните** масиви. Ще разгле­даме различни начини за обхождане на масив, четене от стандарт­ния вход и отпечатване на стандартния изход. Ще дадем много примери за задачи, които се решават с масиви и ще ви покажем колко полезни са те.

Автор на главата е Христо Германов, а редактор е Радослав Тодоров. Съдържанието на цялата глава е базирано на работата на Мариян Ненчев и Светлин Наков от книгата "Въведение в програмирането с Java".

### Глава 8. Бройни системи

В главата "[Бройни системи](#_Глава_8._Бройни_1)" ще разгледаме начините на рабо­та с различни **бройни системи** и представянето на числата в тях. Повече внимание ще отделим на представянето на числата в **десетична**, **двоична** и **шестнаде­сетична** бройна система, тъй като те се използват много често в компютърната и комуникационната техника и в програмира­нето. Ще обя­сним и начините за **кодиране на числовите данни** в компю­търа и видовете кодове, а именно: прав код, обратен код, допълни­телен код и двоично-десетичен код.

Автор на главата е Теодор Божиков, а редактор е Михаил Стойнов. Съдър­жанието на цялата глава е базирано на работата на Петър Велев и Светлин Наков от книгата "Въведение в програмирането с Java".

### Глава 9. Методи

В главата "[Методи](#_Глава_9._Методи)" ще се запознаем подробно с **подпрог­рамите** в програмирането, които в C# се наричат **методи**. Ще обясним кога и защо се използват методи. Ще покажем как се **декларират** методи и какво е **сигнатура** на метод. Ще научим как да създадем собствен метод и съответно как да го използваме (**извикваме**) в последствие. Ще демон­стрираме как можем да използваме **параметри** в методите и как да връщаме резултат от метод. Накрая ще дискутираме някои утвърдени практики при работата с методи. Всичко това ще бъде подкрепено с подробно обяснени **примери** и допъл­нителни задачи.

Автор на главата е Йордан Павлов, а редактори са Радослав Тодоров и Николай Василев. Съдържанието на цялата глава е базирано на работата на Николай Василев от книгата "Въведение в програмирането с Java".

### Глава 10. Рекурсия

В главата "[Рекурсия](#_Глава_10._Рекурсия)" ще се запознаем с **рекурсията** и нейните приложения. Рекурсията представлява мощна техника, при която един **метод извиква сам себе си**. С нея могат да се решават сложни комбина­торни задачи, при които с лекота могат да бъдат изчерпвани различни комбинаторни конфи­гурации. Ще ви покажем много примери за правилно и неправилно **използване на рекурсия** и ще ви убедим колко полезна може да е тя.

Автор на главата е Радослав Иванов, а редактор е Светлин Наков. Съдържанието на цялата глава е базирано на работата на Радослав Иванов и Светлин Наков от книгата "Въведение в програмирането с Java".

### Глава 11. Създаване и използване на обекти

В главата "[Създаване и използване на обекти](#_Глава_11._Създаване)" ще се запоз­наем накратко с основните понятия в обектно-ориентираното програми­ране – **класовете** и **обектите** – и ще обясним как да използваме класовете от стандартните библиотеки на .NET Framework. Ще се спрем на някои често изпол­звани **системни класове** и ще покажем как се създават и използват техни инстанции (обекти). Ще разгледаме как можем да осъщест­вяваме достъп до **полетата** на даден обект, как да извикваме **конструк­тори** и как да работим със **статичните полета** в класовете. Накрая ще обърнем внимание на понятието пространства от имена (**namespaces**), с какво те ни помагат, как да ги включваме и използваме.

Автор на главата е Теодор Стоев, а редактор е Стефан Стаев. Съдържа­нието на цялата глава е базирано на работата на Теодор Стоев и Стефан Стаев от книгата "Въведение в програмирането с Java".

### Глава 12. Обработка на изключения

В главата "[Обработка на изключения](#_Глава_12._Обработка)" ще се запознаем с **изключенията** в обектно-ориентираното програмиране, в частност в езика C#. Ще се научим как да ги прихващаме чрез **конструкцията** try-catch, как да ги предаваме на предходните методи и как да **хвърляме** собствени или прихванати изключения чрез **конструкцията** throw. Ще дадем редица примери за използването им. Ще разгледаме типовете изключения и **йерархията**, която образуват в .NET Framework. Накрая ще се запознаем с предимствата при използването на изключения и с това как най-правилно да ги прилагаме в конкретни ситуации.

Автор на главата е Михаил Стойнов, а редактор е Радослав Кирилов. Съдържанието на цялата глава е базирано на работата на Лъчезар Цеков, Михаил Стойнов и Светлин Наков от книгата "Въведение в програмира­нето с Java".

### Глава 13. Символни низове

В главата "[Символни низове](#_Глава_13._Символни)" ще се запознаем със **сим­волните низове**: как са реализирани те в C# и по какъв начин можем да **обработваме текстово съдържание**. Ще прегледаме различни методи за манипулация на текст; ще научим как да извличаме **поднизове** по зададени параметри, как да **търсим** за ключови думи, както и да **разделяме** един низ по разделители. Ще предоставим полезна информация за **регу­ляр­ните изрази** и ще научим по какъв начин да извличаме данни, отгова­рящи на определен шаблон. Накрая ще се запознаем с методи и класове за по-елегантно и стриктно **формати­ране на текстовото съдържание** на конзолата и с различни методики за извеждане на числа и дати.

Автор на главата е Веселин Георгиев, а редактор е Радослав Тодоров. Съдържанието на цялата глава е базирано на работата на Марио Пешев от книгата "Въведение в програмирането с Java".

### Глава 14. Дефиниране на класове

В главата "[Дефиниране на класове](#_Глава_14._Дефиниране_1)" ще покажем как можем да **дефинираме собствени класове** и кои са елементите на класовете. Ще се научим да декларираме **полета**, **конструктори** и **свойства** в класовете. Ще припомним какво е метод и ще разширим знанията си за модифи­катори и **нива на достъп** до полетата и методите на класовете. Ще разгле­даме особеностите на конструкторите и подробно ще обясним как обек­тите се съхраняват в динамичната памет и как се инициализират полетата им. Накрая ще обясним какво представляват **статичните елементи** на класа – полета (включително константи), свойства и методи и как да ги ползваме.

Автори на главата са Николай Василев, Мира Бивас и Павлина Хаджиева. Съдържанието на цялата глава е базирано на работата на Николай Василев от книгата "Въведение в програмирането с Java".

### Глава 15. Текстови файлове

В главата "[Текстови файлове](#_Глава_15._Текстови)" ще се запознаем с основните похвати при работа с **текстови файлове** в .NET Framework. Ще разясним какво е това **поток**, за какво служи и как се ползва. Ще обясним какво е текстов файл и как се чете и пише в текстови файлове. Ще демонстри­раме и обясним добрите практики за прихващане и **обра­ботка на изклю­чения** при работата с файлове. Разбира се, всичко това ще онагледим и демон­стрираме на практика с много примери.

Автор на главата е Радослав Кирилов, а редактор е Станислав Златинов. Съдържанието на цялата глава е базирано на работата на Данаил Алексиев от книгата "Въведение в програмирането с Java".

### Глава 16. Линейни структури от данни

В главата "[Линейни структури от данни](#_Глава_16._Линейни)" ще се запознаем с някои от основните представяния на данните в програмирането и с **линейните структури от данни**, тъй като много често, за решаване на дадена задача се нуждаем да работим с **последователност от елементи**. Например, за да прочетем тази книга, трябва да прочетем последователно всяка една страница т.е. да обходим последова­телно всеки един от елементите на множеството от нейните страници. Ще видим как при определена задача една структура е по-ефективна и удобна от друга. Ще разгледаме структурите "**списък**", "**стек**" и "**опашка**" и тяхното приложе­ние. Подробно ще се запознаем и с някои от реализациите на тези структури.

Автор на главата е Цвятко Конов, а редактор е Дилян Димитров. Съдър­жанието на глава е базирано в голяма степен на работата на Цвятко Конов и Светлин Наков от книгата "Въведение в програмирането с Java".

### Глава 17. Дървета и графи

В главата "[Дървета и графи](#_Глава_17._Дървета)" ще разгледаме т. нар. **дърво­видни структури от данни**, каквито са дърветата и графите. Познаването на свойствата на тези стру­ктури е важно за съвременното програмиране. Всяка от тях се използва за моде­лирането на проблеми от реалността, които се решават ефективно с тяхна помощ. Ще разгледаме в детайли какво представляват **дърво­видните структури от данни** и ще покажем техните основни предимства и недос­татъци. Ще дадем примерни реализа­ции и задачи, демонстриращи реалната им употреба. Ще се спрем по-подробно на **двоичните дървета**, наредените двоични дървета за претър­сване и балансираните дървета. Ще разгледаме структу­рата от данни "**граф**", видовете графи и тяхната употреба. Ще покажем и къде в .NET Framework се използват имплемен­тации на балансирани дървета за търсене.

Автор на главата е Веселин Колев, а редактор е Илиян Мурданлиев. Съдър­жанието на цялата глава е базирано на работата на Веселин Колев от книгата "Въведение в програмирането с Java".

### Глава 18. Речници, хеш-таблици и множества

В главата "[Речници, хеш-таблици и множества](#_Глава_18._Речници,)" ще разгле­даме някои по-сложни структури от данни като **речници** и **множества**, и техните реализации с **хеш-таблици** и **балан­сирани дървета**. Ще обясним в детайли какво представляват хеширането и хеш-таблиците и защо са толкова важни в програмирането. Ще дискути­раме понятието "**колизия**" и как се полу­чават колизиите при реализация на хеш-таблици и ще пред­ложим различни подходи за разрешаването им. Ще разгледаме абстракт­ната структура данни "множество" и ще обясним как може да се реализира чрез **речник** и чрез **балансирано дърво**. Ще дадем при­мери, които илю­стрират приложението на описаните структури от данни в практиката.

Автор на главата е Михаил Вълков, а редактор е Цвятко Конов. Съдържа­нието на цялата глава е частично базирано на работата на Владимир Цанев от книгата "Въведение в програмирането с Java".

### Глава 19. Структури от данни – съпоставка и препоръки

В главата "[Структури от данни – съпоставка и препоръки](#_Глава_19._Структури)" ще **съпоставим една с друга структурите данни**, които се разглеждат в предходните глави, по отношение на **скоростта**, с която извършват основните опера­ции (добавяне, търсене, изтриване и т.н.). Ще дадем конкретни препо­ръки в какви ситуации **какви структури от данни да ползваме**. Ще обясним кога да предпочетем хеш-таблица, кога масив, кога динамичен масив, кога множество, реализирано чрез хеш-таблица и кога баланси­рано дърво. Всички тези структури имат вградена импле­мен­тация в .NET Framework. От нас се очаква единствено да се научим да преценя­ваме **кога коя струк­тура да ползваме**, за да пишем ефективен и надежден програмен код.

Автори на главата са Николай Недялков и Светлин Наков, а редактор е Веселин Колев. Съдържанието на цялата глава е базирано на работата на Светлин Наков и Николай Недялков от книгата "Въведение в програмира­нето с Java".

### Глава 20. Принципи на обектно-ориентираното програмиране

В главата "[Принципи на обектно-ориентираното програмиране](#_Глава_20._Принципи)" ще се запознаем с принципите на обектно-ориен­тираното програми­ране: **насле­дяване** на класове и имплементиране на **интерфейси**, **абстрак­ция** на данните и на поведението, **капсулация** на данните и скриване на информация за имплементацията на класовете, **полиморфизъм** **и вирту­ални методи**. Ще обясним в детайли принципите за свързаност на отговор­ностите и функционално обвързване (**cohesion** и **coupling**). Ще опишем накратко как се извършва обектно-ориентирано моделиране и как се създава обектен модел по описание на даден бизнес проблем. Ще се запознаем с езика **UML** и ролята му в процеса на обектно-ориентираното моделиране. Накрая ще разгледаме съвсем накратко концепцията "**шаб­лони за дизайн**" и ще дадем няколко типични примера за шаблони, широко използвани в практиката.

Автор на главата е Михаил Стойнов, а редактор е Михаил Вълков. Съдър­жанието на цялата глава е базирано на работата на Михаил Стойнов от книгата "Въведение в програмирането с Java".

### Глава 21. Качествен програмен код

В главата "[Качествен програмен код](#_Глава_21._Качествен)" ще разгледаме основ­ните правила за писане на **качествен програмен код**. Ще бъде обърнато внимание на **именуването** на елементите от програмата (променливи, методи, класове и други), прави­лата за **форматиране** и подреждане на кода, добрите практики за изграж­дане на висококачествени **класове** и **методи** и принципите за качествена **докумен­тация** на кода. Ще бъдат дадени много примери за качествен и некаче­ствен код. В процеса на работа ще бъде обяснено как да се използва средата за програмиране, за да се автомати­зират някои операции като форма­тиране и **преработка на съществуващ код**, когато се налага.

Автор на главата е Михаил Стойнов, а редактор е Павел Дончев. Съдържа­нието на цялата глава е базирано частично на работата на Михаил Стойнов, Светлин Наков и Николай Василев от книгата "Въведение в програмирането с Java".

### Глава 22. Ламбда изрази и LINQ заявки

В главата "[Ламбда изрази и LINQ заявки](#_Глава_22._Ламбда)" ще се запознаем с част от по-сложните възможности на езика C# и по-специално ще разгле­даме как се правят **заявки към колекции** чрез **ламбда изрази** и LINQ заявки. Ще обясним как да добавяме функционалност към съществуващи вече класове, използвайки разширя­ващи методи (**extension methods**). Ще се запознаем с **анонимните типове** (anonymous types), ще опишем накратко какво представляват и как се използват. Ще разгледаме **ламбда изразите** (lambda expressions) и ще покажем с примери как работят повечето вградени **ламбда функции**. След това ще обърнем по-голямо внимание на езика за заявки LINQ, който е част от C#. Ще научим какво представлява, как работи и какви заявки можем да конструираме с него. Накрая ще се запознаем с ключовите думи за езика **LINQ**, тяхното значение и ще ги демонстрираме, чрез голям брой примери.

Автор на главата е Николай Костов, а редактор е Веселин Колев.

### Глава 23. Как да решаваме задачи по програмиране?

В главата "[Как да решаваме задачи по програмиране?](#_Глава_22._Как)" ще дискутираме един препоръчителен подход за **решаване на задачи по прог­рамиране** и ще го илюстрираме нагледно с реални примери. Ще дискути­раме инженерните **принципи**, които трябва да следваме при **решаването на задачи** (които важат в голяма степен и за задачи по математика, физика и други дисциплини) и ще ги покажем в действие. Ще опишем **стъпките**, през които преминаваме при решаването на няколко примерни задачи и ще демонстрираме какви грешки се получават, ако не следваме тези стъпки. Ще обърнем внимание на някои **важни стъпки при решаването на задачи** (като например тестване), които обикновено се пропускат.

Автор на главата е Светлин Наков, а редактор е Веселин Георгиев. Съдържанието на цялата глава е базирано изцяло на работата на Светлин Наков от книгата "Въведение в програмирането с Java".

### Глави 24, 25, 26. Практически задачи за изпит по програмиране

В главите "[Практически задачи за изпит по програмиране](#_Глава_24._Практически)" ще разгледаме условията и ще предложим решения на **девет примерни задачи** от три примерни изпита по програмиране. При решаването им ще приложим на практика описаната **методология** в главата "[Как да решаваме задачи по програмиране](#_Глава_22._Как)".

Автори на главите са съответно Стефан Стаев, Йосиф Йосифов и Теодор Стоев, а редактори са съответно Радослав Тодоров, Радослав Иванов и Йосиф Йосифов. Съдържанието на тези глави е базирано в голяма степен на работата на Стефан Стаев, Светлин Наков, Радослав Иванов и Теодор Стоев от книгата "Въведение в програмирането с Java".

## За използваната терминология

Тъй като настоящият текст е на **български език**, ще се опитаме да ограни­чим употребата на английски термини, доколкото е възможно. Съществу­ват обаче основателни причини да използваме и **англий­ските тер­мини** наред с българските им еквиваленти:

* По-голямата част от техническата документация за C# и .NET Framework е на **английски език** (повечето книги и в частност официалната докумен­тация) и затова е много важно читателите да знаят английския еквивалент на всеки използван **термин**.
* Много от използваните **термини** не са пряко свързани със C# и са навлезли отдавна в програмисткия жаргон от английски език (напри­мер "**дебъгвам**", "**компилирам**" и "**плъгин**"). Тези термини ще бъдат изписвани най-често на кирилица.
* Някои термини (например "framework" и "deployment") са **трудно преводими** и трябва да се използват заедно с оригинала в скобки. В настоящата книга на места такива термини са превеждани по раз­лични начини (според контекста), но винаги при първо срещане се дава и оригинални­ят термин на английски език.

## Как възникна тази книга?

Често се случва някой да попита **от коя книга да започне да се учи на програмиране**. Срещат се ентусиазирани младежи, които искат да се учат да програмират, но не знаят от къде да започнат. За съжаление е трудно да им бъде препоръчана добра книга. Можем да се сетим за много книги за C# – и на български и на английски, но никоя от тях **не учи на програ­миране**. Няма много книги (особено на български език), които да учат на **концепциите** на програмирането, на **алгоритмично мислене**, на **структури от данни**. Има книги за начинаещи, които учат на езика C#, но не и на основите на прог­рамирането. Има и няколко хубави книги за програми­ране на бъл­гарски език, но са вече остарели и учат на отпаднали при еволюцията езици и технологии. Известни са няколко такива книги за C и Паскал, но не и за C# или Java. В крайна сметка е трудно да се сетим за **хубава книга**, която горещо да препоръчаме на всеки, който иска да се захване с програ­миране от нулата.

Липсата на **хубава книга по програмиране** за начинаещи в един момент мотивира главния организатор на проекта [**Светлин Наков**](http://www.nakov.com) да събере автор­ски колектив, който да се захване и **да напише** най-накрая такава книга. Решихме, че можем да помогнем и да дадем знания и вдъхновение на много млади хора да се захванат сериозно с програмиране.

### Историята на тази книга

Тази книга възникна като превод и адаптация на книгата "[Въведение в програмирането с Java](http://www.introprogramming.info/intro-java-book/)" към C# и .NET Framework и съответно наследява историята на своя предшественик като добавя нови нотки на нововъве­дения, изменения и допълнения от новия авторски колектив.

Историята на книгата "[Въведение в програмирането с Java](http://www.introprogramming.info/intro-java-book/)" е дълга и интересна. Тя започва с въвеждащите курсовете по програмиране в **Национална академия по разработка на софтуер** (НАРС) през 2005 г., когато под ръководството на Светлин Наков за тях е изготвено учебно съдържание за курс "**Въведение в програмира­нето със C#**". След това то е адаптирано към Java и така се получава курсът "Въведение в програми­рането с Java". През годините това учебно съдържание претърпява доста промени и подобрения и достига до един изчистен и завършен вид.

### Събиране на авторския екип

Работата по оригиналната книга "[Въведение в програмирането с Java](http://www.introprogramming.info/intro-java-book/)" започва в един топъл летен ден (август 2008 г.), когато основният автор **Светлин Наков**, вдъхновен от идеята за **написване на учебник** за курсовете по "Въведение в програмирането" събира екип от двадесетина млади софту­ерни инже­нери, ентусиасти, които имат желание да споделят знанията си и да напишат по една глава от книгата.

Светлин Наков дефинира **учебното съдържание** и го разделя в глави и създава **шаблон** за съдържанието на всяка глава. Шаблонът съдържа структурата на текста – всички основни заглавия в дадената глава и всички подзаглавия. Остава да се напишат текста, примерите и задачите.

На първата **среща на екипа** учебното съдържание претърпява малко промени. По-обемните глави се разделят на няколко отделни части (например структурите от данни), възникват няколко нови глави (напри­мер работа с изключения) и се определят **автори** и **редактори** за всяка глава. Идеята е проста: всеки да напише по 1 глава от книгата и накрая да бъдат съединени в книга. За да няма голяма разлика в стиловете, форматирането и начина на представяне на информацията авторите приемат единно **ръководство на писателя**, в което строго се описват всички правила за писане. В крайна сметка всеки си има тема и писането започва.

За проекта се създава сайт за съвместна работа в екип в Google Code на адрес <http://code.google.com/p/introjavabook/>, където стои последната версия на всички текстове и материали по книгата.

### Задачите и сроковете

Както във всеки проект, след разпределяне на **задачите** се слагат **крайни срокове** за всяка от тях, за да се планира работата във времето. По план книгата трябва да излезе от печат през октомври 2008 г., но това не се случва в срок, защото много от авторите се забавят, а някои въобще не изпълняват поетия ангажимент.

Когато идва първия краен срок едва **половината от авторите са готови** на време. Сроковете се удължават и голяма част от авторите завършват работата по своята глава. Започва работата на **редакторите**. Паралелно някои автори дописват. За някои глави се търсят нови автори, защото оригиналният автор се проваля и бива отстранен.

Няколко месеца по-късно книгата е готова на 90%, авторите загубват ентусиазъм работата започва да върви много бавно и мъчно. Светлин Наков се опитва да компенсира и да **дописва недовършените теми**, но работата е много. Въпреки 30-те часа, които той влага като труд всеки свободен уикенд, работата е много и все не свършва месеци наред.

**Всички автори подценяват сериозно обема на работата** и това е основно причината за забавянето на нейната поява. Авторите си мислят, че писането става бързо, но истината е, че за **една страница текст** (четене, писане, редактиране, преправяне и т.н.) отиват средно по **1-2 часа работа**, та дори и повече. Сумарно за написването на цялата книга са вложени около **800-1000 работни часа труд**, разпределени сред всички автори и редак­тори, което се равнява на над **6 месеца работа** на един автор на пълен работен ден. Понеже всички автори пишат в свободното си време, рабо­тата върви бавно и отнема 4-5 месеца.

Книгата "**Въведение в програмирането с Java**" излиза официално през **януари 2009 г.** и се разпространява безплатно от официалния й уеб сайт: [www.introprogramming.info](http://www.introprogramming.info).

### Превеждане на книгата към C#

[Книгата "Въведение в програмирането с Java"](http://www.introprogramming.info) се чете с голям интерес. Към декември 2009 г. тя е изтеглена над 6 000 пъти и първият тираж на хартия е почти изчерпан, а сайтът й е посетен над 50 пъти на ден.

През ноември 2009 г. стартира проект за **"превеждане" на книгата "Въве­дение в програмирането с Java" към C#** под заглавие "Въведение в прог­рамирането със C#". Събира се отново голям екип от софтуерни инженери под ръководството на **Светлин Наков** и **Веселин Колев**. Идеята е да се **адаптира текста на книгата**, заедно с всички примери, демон­страции, обяснения към тях, задачи, решения, упражнения и упътвания към C# и .NET Framework. Работата изглежда, че не е много – трябва да се прочете внимателно текста, да се адаптира за C#, да се преработят всички примери и да се заменят всички класове, методи и технологии, свързани с Java със съответните им C# класове, методи и технологии. Лесна на пръв поглед задача, която обаче се оказва **времеотнемаща**. Както може да се очаква, при проекти, които се разработват от широк колектив автори, в свободно им време и на добра воля, книгата е завършена за около половин година. Тогава излиза **предварителната версия** на книгата, в която са открити доста грешки и неточности. За да бъдат изчистени, екипът работи още около година и успява да изглади текста, примерите и задачите за упражнения до вид, подходящ за официално издаване на хартия. Някои от главите се налага да бъдат сериозно редактирани, почти пренаписани, добавя се и главата за ламбда изрази и LINQ.

Новият проект също е с **отворен код** и работата по него е публично дос­тъпна в Google Code: <http://code.google.com/p/introcsharpbook/>. Книгата остава със същия брой глави и няма сериозни промени по същество. За **автори** и **редактори** са поканени всички оригинални автори на съответ­ните глави от книгата "Въведение в програмирането с Java", но повечето от тях се отказват и към екипа се присъединяват **много нови автори**. В крайна сметка проектът завършва с успех и [книгата "Въведение в програ­мирането със C#"](http://www.introprogramming.info) излиза през **лятото на 2011 г.** Сайтът на новата книга е същият ([introprogramming.info](http://www.introprogramming.info)), като е разделен на секция за C# и Java.

Част от авторите проявяват интерес за адаптиране на книгата още веднъж, към **езика C++**, но не е твърдо решено ще бъде ли стартиран такъв проект и евентуално кога. Появяват се и идеи за превод на книгата на **английски език**, но за такава амбициозна задача Светлин Наков намира ресурс едва през 2012 г. и след година усилена работа се появява **английската версия** на настоящата книга – "[Fundamentals of Computer Programming with C#](http://www.introprogramming.info/english-intro-csharp-book/)".

## Авторският колектив

Авторският колектив (на старата и на новата книга) е наистина главният виновник за съществуването на тази книга. Написването на текст с такъв обем и такова качество е **сериозна задача**, която изисква **много време**.

Идеята за участие на толкова много автори е добре проверена, тъй като по подобен начин са написани вече **няколко други книги** (като ["Програ­миране за .NET Framework" – част 1 и 2](http://www.devbg.org/dotnetbook/)). Въпреки че отделните глави от книгата са писани от различни автори, те следват **единен стил и високо качество** (макар и не еднакво във всички глави). Текстът е добре структу­риран, с много заглавия и подза­главия, с много и подходящи примери, с добър стил на изказ и еднакво форма­тиране.

Екипът, написал настоящата книга, е съставен от хора, които имат силен интерес към програмирането и желаят **безвъзмездно да споделят своите знания** като участват в написването на една или няколко от главите. Най-хубавото е, че всички автори, съавтори и редак­тори от екипа по разра­ботката на книгата са действащи **програмисти с реален практически опит**, което означава, че читателят ще почерпи знания, практики и съвети от хора, реализирали се в софтуерната индустрия.

Участниците в проекта дадоха своя труд **безвъзмездно**, без да получат материални или други директни облаги, защото подкрепяха идеята за написване на добра книга за начинаещи програмисти на български език и имаха силно желание да помогнат на своите бъдещи колеги да навлязат бързо в програмирането.

Следва кратко представяне на **авторите на книгата** "Въведение в програ­мирането със C#" (по азбучен ред). Оригиналните автори на съответните глави от книгата "Въведение в програмирането с Java" също са упоменати по подходящ начин, тъй като техните заслуги в някои глави са по-големи, отколкото заслугите на следващите автори след тях, които са адаптирали текста и примерите към C#.

### Веселин Георгиев

Веселин Георгиев е съосновател на **Lead IT** и софтуерен разработчик в **Abilitics** ([www.abilitics.com](http://www.abilitics.com)). Завършил е магис­тър "Електронен бизнес и Електронно управление" в Софийски Универ­ситет "Св. Климент Охридски", след бакалавърска степен по Информатика също в Софийски Университет.

Веселин е **Microsoft Certified Trainer**. Бил е лектор в конференциите "Дни на Майкрософт" през 2011 и 2009 г.. Участва като преподавател в курсовете "Програмиране с .NET & WPF" и "Разработка на богати интернет приложения (RIA) със Silverlight" в Софийски Университет. Опитен лектор, работил върху обучението на софтуерни специалисти за практическа работа в ИТ индустрията.

Професионалните му интереси са насочени към .NET обучения, разработ-ката на разнообразни .NET приложения, софтуерни архитектури. Сертифи­циран е като **Microsoft Certified Professional Developer**.

Можете да се свържете с Веселин в Twitter: [twitter.com/VeselinGeorgiev](mailto:twitter.com/VeselinGeorgiev) или по e-mail: [veselin.vgeorgiev@gmail.com](mailto:veselin.vgeorgiev@gmail.com).

### Веселин Колев

**Веселин (Веско) Колев** е водещ софтуерен инженер с повече от 10 години професионален опит. Той е работил с различни компании, в които е **ръководил разработката** на разнообразни софтуерни проекти и екипи. Като ученик е участвал в редица **състезания** по математика, информа­тика и информационни технологии, в които е заемал престижни места. Притежава **бакалавърска степен** по "Компютърни науки" от "Факултета по математика и информатика" на СУ "Св. Климент Охридски".

Веско е опитен **лектор**, работил върху обучението на софтуерни специалисти за практическа работа в ИТ индустрията. Участва като **преподавател** във Факултета по математика и информатика на Софийски университет, където е водил курсовете "Съвременни Java технологии" и "Качествен програмен код". Водил е аналогични обучения и в Технически университет – София.

Основните **интереси** на Веско включват дизайн на софтуерни системи, качествен програмен код, работа с legacy код, и управление и развитие на големи екипи. Проектите, по които е работил, включват големи уеб базирани и декстоп системи, мобилни приложения, OCR, системи за машинен превод, бизнес софтуер и много други. Веско е съавтор и в **книгата "Въведение в програмирането с Java"**.

В момента Веско заема длъжността "**директор, разработка на софтуер**" в софтуерната компания **Телерик**, оглавявайки дивизията Business Services. Дивизията се състои от екипи, отговарящи за разработ­ката на критични бизнес системи, върху които оперира бизнесът на Телерик, както и за корпоративния уеб сайт на компанията ([www.telerik.com](http://www.telerik.com/)).

Част от своя ежедневен си опит Веско Колев споделя онлайн в личния си уеб сайт [http://veskokolev.com](http://veskokolev.com/), както и в профила си на разработчик в GitHub: <https://github.com/veskokolev>.

Можете да се свържете с Веско в Twitter (<https://twitter.com/veskokolev>), в LinkedIn (<https://www.linkedin.com/in/veselinkolev>) или чрез личния му e-mail: [vesko.kolev@gmail.com](mailto:vesko.kolev@gmail.com).

### Дилян Димитров

Дилян Димитров е **сертифициран софтуерен разработчик** с професио­нален опит в изграждането на средни и големи уеб базирани системи върху .NET платформата. Интересите му включват разработка, както на уеб, така и на десктоп приложения с последните **технологии на Microsoft**. Той е завършил Факултета по математика и информатика на Софийския университет "Св. Климент Охридски" със специалност "Информатика". Може да се свържете с него по e-mail: [dimitrov.dilqn@gmail.com](mailto:dimitrov.dilqn@gmail.com) или да посетите личният му блог на адрес: <http://dilyandimitrov.blogspot.com>.

### Илиян Мурданлиев

Илиян Мурданлиев е **софтуерен разработчик** във фирма Ниърсофт ([www.nearsoft.eu](http://www.nearsoft.eu)). Завършил е магистър "Компютърни Технологии и Приложно Програмиране" в Технически Университет - София. Бакалавър е в същия университет в специалност "Приложна Математика". Завършил е езикова гимназия с английски език.

Илиян е участвал в сериозни проекти при разработката както на front-end визуализацията, така и на back-end логиката. Подготвял е и е водил **обучения по C#** и други езици за програмиране. Интересите на Илиян са в областта на новите технологии свързани с .NET, Windows Forms и Web базираните технологии, шаблони за дизайн, алгоритми и софтуерно инженерство. Обича разчупени проекти, в които не трябват само познания, но и логическо мислене.

Личният му блог е достъпен на адрес: <http://imurdanliev.wordpress.com>. Можете да се свържете с него по e-mail: [i.murdanliev@gmail.com](mailto:i.murdanliev@gmail.com).

### Йосиф Йосифов

Йосиф Йосифов е **софтуерен разработчик** в Telerik ([www.telerik.com](http://www.telerik.com)). Интересите му са свързани предимно с .NET технологиите, шаблоните за дизайн и компютърните алгоритми. Участвал е в множество **състезания** и олимпиади по информатика. В момента той следва Компютърни науки във Факултета по математика и информатика на Софийски Университет "Св. Климент Охридски".

Личният блог на Йосиф е достъпен от адрес: <http://yyosifov.blogspot.com>. Можете да се свържете с него по e-mail: [cypressx@gmail.com](mailto:cypressx@gmail.com).

### Йордан Павлов

Йордан Павлов е завършил бакалавърска и магистърска степен, специал­ност "Компютърни системи и технологии" в Технически университет – София. Той е **софтуерен разработчик** в Телерик ([www.telerik.com](http://www.telerik.com)) със значителен опит в разработката на софтуерни компоненти.

Интересите му са най-вече в следните области: обектно-ориентиран дизайн, шаблони за дизайн, разработка на качествен софтуер, географски информационни системи (ГИС), паралелна обработка и високо производи­телни системи, изкуствен интелект, управление на екипи.

Йордан е победител на локалните финали за България на състезанието **Imagine Cup 2008** в категория "Софтуерен дизайн" както и на световните финали в Париж, където печели престижната награда на Microsoft "**The Engineering Excellence Achievement Award**". Работил е заедно с инженери на Майкрософт в централата на компанията в Редмънд, САЩ, където е натрупал полезни знания и умения за разработката на сложни софтуерни системи.

Йордан е и носител на златен знак за "принос към младежкото иноваци­онно и информационно общество". Участвал е в множество **състезания и олимпиади** по информатика.

Личният му блог е достъпен на адрес <http://yordanpavlov.blogspot.com>. Можете да се свържете с него по e-mail: [iordanpavlov@gmail.com](mailto:iordanpavlov@gmail.com).

### Мира Бивас

Мира Бивас е ентусиазиран млад **програмист** в един от ASP.NET екипите на Тelerik ([www.telerik.com](http://www.telerik.com)). Тя е студентка във Факултета по математика и информатика на Софийски университет "Св. Климент Охридски", специ­алност "Приложна математика". Мира е завършила Intro C# и Core .NET курсовете на Национална академия по разработка на софтуер (НАРС).

Може да се свържете с нея по e-mail: [mira.bivas@gmail.com](mailto:mira.bivas@gmail.com).

### Михаил Вълков

Михаил Вълков е софтуерен разработчик от 2000 г. През годините Михаил се е сблъсквал с множество технологии и платформи за разработка, сред които Microsoft .NET, ASP, Delphi. От 2004 г. Михаил **разработва софтуер във фирма Телерик** ([www.telerik.com](http://www.telerik.com)). Там той участва в изграждането на редица компоненти за ASP.NET, Windows Forms, Silverlight и WPF. През последните години, Михаил ръководи едни от най-добре развиващите се екипи в компанията.

Можете да се свържете с него чрез e-mail: [m.valkov@gmail.com](mailto:m.valkov@gmail.com).

### Михаил Стойнов

Михаил Стойнов е магистър по "Стопанско управление" в Софийски уни­верситет. Бакалавърската си степен по "Информатика" е завършил отново в Софийски Университет. Ръководил е развой­ната дейност в Матерна България ([www.materna.bg](http://www.materna.bg)) и компания за **информационна сигурност**.

Михаил е професионален **разработчик на софтуер**, консултант и пре­подавател с дългогодишен опит. От няколко години той е хоноруван **преподавател** във Факултета по математика и информатика като досега е водил лекции в курсовете "Теория на мрежите", "Програмиране за .NET Framework", "Разработка на Java уеб приложения", "Шаблони за дизайн" и "Качествен програмен код". Преподавал е и в Нов български университет.

Той е **автор** на редица статии и публикации и **лектор** на множество конференции и семинари в областта на софтуерните технологии и информационната сигурност. Михаил е участвал като съавтор в книгите "[Програмиране за .NET Framework](http://www.devbg.org/dotnetbook/)" и "[Въведение в програмирането с Java](http://www.introprogramming.info/intro-java-book/)". Участвал е в академичната програма на Microsoft - MSDN Academic Alliance и е бил лектор в академичните дни на Майкрософт.

Михаил е **водил IT обучения** в България и в чужбина. Бил е лектор на курсове по Java, Java EE, SOA, Spring в Национална академия по разработка на софтуер (НАРС). Член е на Българската асоциация на разработчиците на софтуер (БАРС).

Михаил е работил в международните офиси на Siemens, HP, EDS в Холандия и Германия, където е натрупал **сериозен опит** както за софтуер­ното изкуство, така и за качественото писане на софтуер чрез участието си в големи софтуерни проекти. Неговите интереси обхващат информационна сигурност, изгражда­не на софтуерни архитектури и дизайн, B2B интегри­ране на разнородни информационни системи, оптимизация на бизнес процеси и софтуерни системи основно върху платформите Java и .NET. Михаил е участвал в **десетки софтуерни проекти** и има значителен опит с уеб приложения и уеб услуги, разпределени системи, релационни бази от данни и ORM инструменти и управлението на проекти и екипи за разработка на софтуер.

Личният му **блог** е достъпен на адрес: [http://mihail.stoynov.com](http://mihail.stoynov.com/).

### Николай Василев

Николай Василев е завършил бакалавърската си степен във Факултета по математика и информатика на Софийски университет "Св. Кл. Охридски", специалност "Математика и информатика". Има магистърска степен от университе­та в Малага, Испания, специалност "**Софтуерно инженерство и изкуствен интелект**". Завършил е магистърската програма на Софийски университет "Св. Кл. Охридски", специалност "Уравнения на математичес­ката физика и приложения".

Той е професионален **разработчик на софтуер**, като е работил, както в български, така и международни компании.

Съавтор е на книгата "[Въведение в програмирането с Java](http://www.introprogramming.info/intro-java-book/)".

В периода 2002-2005 г е водил упражненията към курсовете по програми­ране водени от доц. Божидар Сендов, "Увод в програмирането" и "Структури от данни и програмиране".

Интересите му са свързани, както със **софтуерната индустрия** – проекти­ране, имплементация и интегра­ция­ на софтуерни системи (предимно върху платформата Java), така и с участие в академични и научно­изследо­вателски дейности в областите на софтуерното инженерство, изкуствения интелект и механиката на флуи­дите.

Участвал е в множество разнородни проекти и има опит в разработката на уеб приложения и уеб услуги, релационни бази от данни и ORM платфор­ми, модулно програмиране с OSGI, потребителски интерфейси, разпреде­лени и VOD системи.

Личният блог на Николай Василев е на адрес: [http://blog.nvasilev.com](http://blog.nvasilev.com/)

### Николай Костов

Николай Костов работи като **technical trainer** в отдел "технологично обучение" в Телерик. Занимава се с обученията в **софтуерната академия на Телерик** ([http://academy.telerik.com](http://academy.telerik.com/)). Учил е във Факултета по матема­тика и информатика на Софийския университет "Св. Климент Охридски", специалност "Компютърни науки". Сертифициран е като **Microsoft Certified Trainer**.

Николай е дългогодишен участник в редица ученически и студентски **олимпиади и състезания по информатика**. Двукратен победител в проект­ните категории "Приложни програми" и "Интернет приложения" на нацио­налната олимпиадата по информационни технологии. Има богат опит в проектирането и изграждането на интернет приложения, алгоритмичното програмиране и обработката на големи обеми данни.

Основните му интереси са свързани с разработването на софтуерни при­ложения, алгоритми, структури от данни, всичко свързано с .NET техноло­гиите, сигурност на уеб приложенията, автоматизиране на обработката на данни, web crawlers и др.

Личният **блог** на Николай е достъпен на адрес: [http://nikolay.it](http://nikolay.it/). Профилът му в **GitHub** е достъпен от: <https://github.com/NikolayIT/>.

### Николай Недялков

Николай Недялков е президент на [Асоциация за информационна сигурност](http://www.iseca.org/). Работил е като технически директор на портала за електронни разплаща­ния и услуги **eBG.bg**, изпълнителен директор на [**Информационно обслуване АД**](http://www.is-bg.net), консултант по електорнно управление и бизнес съветник и консултант в други компании. Николай е **професионален разработчик на софтуер**, консултант и **преподавател** с дългогодишен опит. Той е автор на редица статии и публикации и **лектор** на множество конференции и семинари в областта на софтуерните техно­логии и информационната сигурност. Преподавателският му опит се прос­тира от асистент по "Структури от данни в програмирането", "Обектно-ориентирано програми­ране със C++" и "Visual C++" до лектор в курсовете "Мрежова сигурност", "Сигурен програмен код", "[Интернет програмиране с Java](http://www.nakov.com/inetjava/)", "[Конструиране на качествен програмен код](http://codecourse.sourceforge.net/)", "[Програмиране за платформа .NET](http://www.nakov.com/dotnet/2003/)" и "Разработка на приложения с Java".

Николай Недялков е инициатор на проекта **InfoStart** за обучение на софтуерни инженери и предприемачество в ИТ сектора: <https://infostart.eu>.

Интересите на Николай са концентри­рани върху **изграждането и управле­нието на инфор­мационни и комуника­ционни решения**, моде­лирането и управлението на бизнес процеси в големи организации и в държавната администрация. Николай има бакалавърска и магистърска степен от Факултета по мате­матика и информатика на Софийски университет "Св. Климент Охридски". Като ученик е дългогодишен **състезател по програми­ране**, с редица при­зови отличия.

Профил в LinkedIn: <https://www.linkedin.com/in/nnedyalkov>.

### Павел Дончев

Павел Дончев **основател на фирма** [**eVeliko**](http://eveliko.com) във Велико Търново, където разработва уеб проекти и системи за управление на съдръжание. Преди това е работил като **програмист** във фирма Тelerik ([www.telerik.com)и](http://www.telerik.com)и) се е занимавал с разработката на уеб приложения, предимно за вътрешни нуж­ди на фирмата. Учил е задочно теоретична физика в Софийски универ­ситет "Св. Климент Охридски". Занимавал се е с разработка на Windows и Web приложения в различни сектори на бизнеса – ипотечни кредити, онлайн магазини, автоматика, Web UML диаграми. Интересите му са предимно в сферата на автоматизирането на процеси с технологиите на Майкрософт.

Личният му блог е достъпен от адрес <http://donchevp.blogspot.com>.

### Павлина Хаджиева

Павлина Хаджиева е **програмист** във фирма Телерик ([www.telerik.com](http://www.telerik.com)). Завършила е магистър "Разпределени системи и мобилни технологии" във Факултета по математика и информатика на Софийски Университет "Св. Климент Охридски". Бакалавърската си степен по "Химия и Информатика" е завършила също в Софийски Университет.

Професионалните й интереси са насочени към **уеб технологиите**, в част­ност ASP.NET, както и цялостната разработка на приложения, базирани на .NET Framework.

Можете да се свържете с Павлина Хаджиева на нейния e-mail: [pavlina.hadjieva@gmail.com](mailto:pavlina.hadjieva@gmail.com)

### Радослав Иванов

Радослав Иванов е **софтуерен инженер**, работил с широк набор от техно­логии. Завършил е Факултета по математика и информатика на Софийски университет "Св. Климент Охридски" и има **сериозен професионален опит** в разработката на софтуер. Той е **лектор** в редица курсове в Софийски университет "Св. Климент Охридски", частни компании и организации и е **съавтор на книгите** "[Програмиране за .NET Framework](http://www.devbg.org/dotnetbook/)" и "[Въведение в програмирането с Java](http://www.introprogramming.info/intro-java-book/)". Работил е като софтуерен инженер в Европей­ската организация за ядрени изследвания (CERN) – [www.cern.ch](http://www.cern.ch). Сред професионалните му интереси са Java технологиите, .NET платфор­мата, архитектура и дизайн на софтуерни системи и др.

Можете да се свържете с него в LinkedIn: [linkedin.com/in/radoslavivanov](https://www.linkedin.com/in/radoslavivanov).

### Радослав Кирилов

Радослав Кирилов е старши **софтуерен разработчик** във фирма Телерик ([www.telerik.com](http://www.telerik.com)). Завършил е Технически университет – София, специал­ност "Компютърни системи и технологии".

Професионалните му интереси са насочени към **уеб технологиите**, в частност ASP.NET, както и цялост­ната разработка на приложения, базирани на .NET Framework.

Радослав е опитен **лектор**, участвал както в провеждането, така и в разработката на материали (презентации, примери, упражнения) за множество курсове в Национална академия по разработка на софтуер (НАРС). Радослав е участвал в преподавателския екип на курса "**Качествен програмен код**" през 2010 година в Технически университет – София и в Софийски университет "Св. Климент Охрид­ски".

Неговият технологичен блог, който той поддържа от началото на 2009 година, е достъпен на адрес <http://radoslavkirilov.blogspot.com>. Можете да се свържете с Радослав на e-mail: [radoslav.pkirilov@gmail.com](mailto:radoslav.pkirilov@gmail.com).

### Радослав Тодоров

Радослав Тодоров е **софтуерен разработчик** завършил бакалавърската си степен във Факултета по математика и информатика на Софийски уни­верситет "Св. Климент Охридски" ([www.fmi.uni-sofia.bg](http://www.fmi.uni-sofia.bg)). Магистърското си образование в областта на компютърните науки получава в Датския технически университет в Люнгбю, Дания (<http://www.dtu.dk>).

Радослав **преподава** още като асистент-преподавател в курсове на IT University, Копенхаген, Дания (<http://www.itu.dk>) и участва в изследо­вателска дейност в проекти на университета от магистърското си образование. Той има **богат опит** в проектирането, разработването и поддръжката на големи софтуерни продукти за различни компании. Трудовия му опит протича в няколко фирми в България. Към настоящия момент работи като софтуерен инженер за Canon Handy Terminal Solutions Europe в Дания ([www.canon-europe.com/Handy\_Terminal\_Solutions/](http://www.canon-europe.com/Handy_Terminal_Solutions/)).

Интересите на Радослав са насочени както към **софтуерните технологии** с езици от високо ниво, така и към продукти, интегриращи цялостни хардуерни и софтуерни решения в индустриалния и частния сектор.

Може да се свържете с Радослав по e-mail: [radoslav\_todorov@hotmail.com](mailto:radoslav_todorov@hotmail.com).

### Светлин Наков

Светлин Наков е основател и **вдъхновител на Софтуерния университет** ([СофтУни](http://softuni.bg)), където обучава хиляди софтуерни инженери и им дава професия и работа в ИТ индустрията.

Наков неспирно преподава **програмиране и софтуерни технологии** през последните 20 години в Софийски университет, НБУ, Технически универ­ситет, НАРС, Telerik Academy, СофтУни и други учебни центрове. През негови курсове и инициативи са преминали десетки хиляди млади хора.

Той е завършил бакалавърска степен по информатика и магистърска степен по разпределени системи и мобилни технологии в Софийски уни­верситет "Св. Климент Охридски". По-късно получава и **докторска степен (PhD) по компютърни науки** с дисертация в областта на изчислителната лингвис­тика, защитена пред Висшата атестационна комисия (ВАК) към Българ­ската академия на науките (БАН).

Неговите интереси обхващат **.NET платформата**, Java технологиите, уеб приложенията, базите данни, уеб услугите, обучението на софтуерни специалисти, информационната сигурност, изграждането на софтуерни архитектури, тех­нологичното предприемачество и управлението на проекти и екипи за разработка на софтуер.

Светлин Наков има **20-годишен опит** като софтуерен инженер, програ­мист, препо­давател и консултант, преминал от Assembler, Basic и Pascal през C и C++ до HTML, CSS, PHP, Java, C# и JavaScript. Участвал е като софтуерен инженер, консултант и ръководител на екипи в десетки проекти за изграждане на информа­ци­онни системи, уеб приложения, системи за управление на бази от данни, бизнес приложения, ERP системи, крипто­графски модули и обучения на софтуерни инженери. На 24 години създава първата си фирма за обучение на софтуерни инже­нери (НАРС), която 5 години по-късно бива погълната от Телерик.

Светлин има сериозен опит в изграждането на учебни материали, курсове и **учебно съдържание**, в подго­товката и провеждането на **курсове за обучения по програмиране и съвре­менни софтуерни технологии**, натрупан по време на преподавателската му практика. Години наред той е хоноруван пре­подавател по съвременни софтуерни технологии във Факултета по математика и информатика на Софийски университет "Св. Климент Охридски" (ФМИ на СУ), Нов Българ­ски университет (НБУ) и Технически университет – София (ТУ-София), където води курсове по "Проектиране и анализ на компютърни алго­ритми", "Интернет програ­миране с Java", "Мрежова сигурност", "Програми­ране за .NET Framework", "Разработка на Java уеб приложения", "Шаблони за дизайн", "Качествен програмен код", "Разработка на уеб приложения с .NET Framework и ASP.NET", "Разработка на Java и Java EE приложения" и "Web Front-End Development" и други (вж. <http://www.nakov.com/courses/>).

Светлин Наков е **участва в създаването на няколко софтуерни академии** за обучение на програмисти и ИТ специалисти – Национална академия по разработка на софтуер (**НАРС**) (2005 – 2009 г.), **Telerik Academy** (2009 – 2013 г.) и [**Софтуерен университет**](http://softuni.bg) (от 2014 г.).

Светлин има десетки **научни и технически публикации**, свързани с раз­работката на софтуер, в български и чуждестранни издания и е водещ автор на книгите "[Програмиране за .NET Framework (том 1 и 2)](http://www.devbg.org/dotnetbook/)", "[Въведе­ние в програмирането с Java](http://www.introprogramming.info/intro-java-book/)", "[Въведе­ние в програмирането със C#](http://www.introprogramming.info/intro-csharp-book/)", "[Fundamentals of Computer Programming with C#](http://www.introprogramming.info/english-intro-csharp-book/)", "[Интернет програмиране с Java](http://www.nakov.com/books/inetjava/)" и "[Java за цифрово под­пис­ване на документи в уеб](http://www.nakov.com/books/signatures/)". Той е редовен лектор на технически конференции, обучения и семинари и до момента е изнесъл **над 100 технически лекции** по различни технологични събития в България и чужбина. Редовен лектор и **вдъхновител** е на конференции за образование, иновации и предприемачество.

Като ученик и студент Светлин е победител в десетки национални състе­зания по програ­миране, **шампион в конкурси и олимпиади** по инфор­матика, програмиране и технологии. Носител е на **4 медала** от междуна­родни олимпи­ади по информа­тика.

През 2004 г. получава **наградата "Джон Атанасов" от Президента на България** за приноса му към развитието на информаци­онните технологии и информа­ционното общест­во.

Той е един от учредителите на Българската асо­циация на разра­ботчиците на софтуер (**БАРС**) ([www.devbg.org](http://www.devbg.org)) и неин председател.

Наков е съучредител и на асоциацията на софтуерните инженери (**ASE**) – [www.ase.bg](http://www.ase.bg).

Неговият **личен уеб сайт** е достъпен от: [www.nakov.com](http://www.nakov.com).

### Станислав Златинов

Станислав Златинов е **софтуерен разработчик** с професионален опит в разработването на уеб и десктоп приложения, базирани на .NET и Java платформите.

Завършил е магистратура по Компютърна мултимедия във Великотърнов­ски университет "Св. Св. Кирил и Методий".

Неговият личен **блог** е достъпен от: [http://encryptedshadow.blogspot.com](http://encryptedshadow.blogspot.com/).

### Стефан Стаев

Стефан Стаев е **софтуерен разработчик**, който се занимава с изграждане на уеб базирани системи на .NET платформата. Професионалните му интереси са свързани с последните .NET технологии, шаблони за дизайн и база от данни. Участник е в авторския екип на книгата "[Въведение в програмирането в Java](http://www.introprogramming.info/intro-java-book/)".

Стефан е завършил бакалавър по "Информатика" във Факултета по математика и информатика на Софийски университет "Св. Климент Охрид­ски". Завършил е "Националната академия по разработка на софтуер" по специалност "Core .NET Developer".

Можете да се свържете с него по e-mail: [stеfosv@gmail.com](mailto:stеfosv@gmail.com). Неговият **Twitter** е на адрес: <http://twitter.com/stefanstaev>.

### Теодор Божиков

Теодор Божиков е **софтуерен разработчик** във фирма Телерик ([telerik.com](http://www.telerik.com)). Завършва магистратурата си по Компютърни системи и технологии в Технически университет – Варна. Освен опита си като програмист в областта на WPF и Silverlight, той е натрупал експертиза и в разработката на ASP.NET уеб приложения. За кратко се занимава с разра­ботката на частни сайтове. В рамките на проектa i-центрове е участвал в изграждането и поддържането на локална мрежа за публично ползване във Фестивалния и конгресен център - Варна. **Водил е курсове** по компю­търна грамотност и основи на компютърните мрежи.

Професионалните интереси на Теодор включват технологии за разработка на уеб и десктоп приложения, архитектури и шаблони за дизайн, мрежи и всякакви нови технологии.

Можете да се свържете с Теодор по e-mail: [t\_bozhikov@yahoo.com](mailto:t_bozhikov@yahoo.com). Неговият **Twitter** е достъпен от: <http://twitter.com/tbozhikov>.

### Теодор Стоев

Теодор Стоев е завършил бакалавърска и магистърска степен по специал­ност Информатика във ФМИ на Софийски университет "Св. Климент Охридски". Магистърската му специализация в СУ е "Софтуерни техно­логии". В момента следва магистърска програма "Computer Science" в Saarland University (Саарбрюкен, Германия).

Теодор е проектант и **разработчик на софтуер** с дългогодишен опит. Участвал е в изграждането на финансови и застрахователни софтуерни системи, редица уеб приложения и корпоративни сайтове. Участвал е активно в разработката на проекта TENCompetence на Европейската комисия. Съавтор е на книгата "[Въведение в програмирането с Java](http://www.introprogramming.info/intro-java-book/)".

Неговите професионални интереси са в областта на обектно-ориентирания анализ, моделиране и изграждане на софтуерни приложения, уеб техно­логиите и в частност изграждането на Rich Internet Applications. Зад гърба си има немалък опит с алгоритмично програмиране: участвал е в редица ученически и студентски национални **състезания по информатика**.

Неговият личен сайт е достъпен от адрес: [http://www.teodorstoev.com](http://www.teodorstoev.com/).

Можете да се свържете с Теодор по e-mail: [teodor.stoev@gmail.com](mailto:teodor.stoev@gmail.com).

### Христо Германов

Христо Германов е **софтуерен инженер**, чиито интереси са свързани предимно с .NET технологиите. Архитектурата и дизайна на уеб базирани системи, алгоритмите и съвременните стандарти за качествен код са също негова страст. Участвал е в разработката както на малки, така и на големи Web и Desktop базирани приложения. Обича предизвикателни задачи и проекти, в които се изисква силно **логическо мислене**. Завършил е специ­алност "Компютърни мрежи" в колеж "Омега" гр. Пловдив и е специали­зирал в "Националната академия по разработка на софтуер", София, по специалност "Core .NET Developer".

Можете да се свържете с него по e-mail: [hristo.germanov@gmail.com](mailto:hristo.germanov@gmail.com).

### Цвятко Конов

Цвятко Конов е **софтуерен разработчик и преподавател** с разностранни интереси и опит. В неговите компетенции влизат области като интеграции на системи, изграждане на софтуерни архитектури, разработване на системи с редица технологии като .NET, ASP.NET, Silverlight, WPF, WCF, RIA, MS SQL Server, Oracle, MySQL, PostgreSQL и PHP.

**Преподавател­ският му опит** включва голяма палитра от курсове – курсове за начинаещи и напреднали върху .NET технологиите, както и специализирани курсове в отделни технологии като ASP.NET, Oracle, .NET Compact Framework, "Каче­ствен програмен код" и други. Цвятко участва в авторския екип на книгата "[Въведение в програмирането в Java](http://www.introprogramming.info/intro-java-book/)". Профе­сионалните му инте­реси включват уеб и десктоп базирани технологии, клиентско ориенти­рани уеб технологии, бази данни и шаблони за дизайн.

Повече информация за него може да намерите на неговия блог: [http://tsvyatkokonov.blogspot.com](http://tsvyatkokonov.blogspot.com/).

## Редакторите

Освен авторите сериозен принос за създаването на книгата имат и редак­торите, които участваха безвъзмездно в проверката на текста и примерите и отстраняването на грешки и други проблеми. Следват техните имена по азбучен ред:

* Веселин Георгиев
* Веселин Колев
* Дилян Димитров
* Дончо Минков
* Илиян Мурданлиев
* Йосиф Йосифов
* Марин Георгиев
* Мира Бивас
* Михаил Вълков
* Михаил Стойнов
* Николай Костов
* Николай Василев
* Павел Дончев
* Радослав Иванов
* Радослав Кирилов
* Радослав Тодоров
* Светлин Наков
* Станислав Златинов
* Стефан Стаев
* Теодор Божиков
* Цвятко Конов

Авторският колектив благодари на Кристина Николова ([krisinikolova.com](http://krisinikolova.com/)) за дизайна на корицата на книгата.

## Книгата е безплатна!

Настоящата книга се разпространява **напълно безплатно** в елек­тронен вид по лиценз, който позволява използването й за всякакви цели, включи­телно и в комерсиални проекти. Книгата се разпространява и в хартиен вид срещу заплащане, което покрива разходите по отпечатването и раз­прост­ранението й, без да се реализира печалба.

## Отзиви

Ако не вярвате напълно на авторския колектив, разработил настоящата книга, може да се вдъхновите от **отзивите** за нея, дадени от водещи световни специалисти, включително софтуерни инженери от Майкрософт.

### Отзив от Никола Михайлов, Microsoft

**Програмирането е яко нещо**! От стотици години хората се опитват да си направят живота по-лесен, за да работят по-малко. Програмирането позволява да се продължи тази тенденция към мързел на човечеството. Ако сте маниак на тема компютри, или просто искате да впечатлите останалите с един хубав сайт или нещо ваше "невиждано досега", добре дошли. Независимо дали сте от сравнително малката група "маниаци", които като видят хубава програма им се завърта главата, или просто искате да се реализирате професионално и да си живеете живота извън работа, **тази книга е за вас**.

Основните принципи на работа на двигател за коли не са се променили с години – гори там нещо (бензин, нафта или каквото сте сипали) и колата върви. По същия начин **основните принципи на програмирането** не са се променили от години насам. Дали ще пишете следващата игра, софтуер за управление на пари в банка или програмирате "мозъка" на новия биоробот, със сигурност ще използвате принципите и структурите от данни, описани в тази книга.

В книгата ще намерите голяма част от **основите на програмирането**. Аналогична фундаментална книга в автомобилната индустрия би била озаглавена "Двигатели с вътрешно горене".

Каквото и да правите, важното е да ви е приятно! Преди да започнете да четете тази книга – **намислете си нещо за програмисти**, което бихте искали да направите – било сайт, игра, или друга програма, която ви харесва! Докато прочитате книгата, мислете кое и как от прочетеното ще използвате за вашата програма! Ако ви е интересно, ще научите и най-сложното нещо с лекота!

**Моята първа програма** (с която се гордея достатъчно, за да говоря публично) беше просто рисуване по екрана със стрелките на клавиа­турата. Доста време ми отне тогава да я направя, но като се получи ми хареса. Пожелавам ви и на вас: да ви харесва всичко свързано с програмирането! Приятно четене на книгата и успешна професионална реализация!

***Никола Михайлов*** *е софтуерен инженер в* ***Майкрософт****, в екипа разработващ Visual Studio. Автор на сайта* [*http://nokola.com*](http://nokola.com)*, лесно се "пали" на тема програмиране; винаги готов когато трябва да се пише нещо добро! Обича да помага на хора с въпроси и желание за програмиране, независимо дали са начинаещи или експерти. При нужда го потърсете по e-mail:* [*nokola@nokola.com*](mailto:nokola@nokola.com)*.*

### Отзив от Васил Бакалов, Microsoft

"Въведение в програмирането със C#" е един смел опит не само да помогне на читателя да направи **първите си стъпки в програмирането**, а също да го запознае с програмната среда и тренира в практическите задачи, които възникват в ежедневието на програмиста. Авторите са намерили добро съчетание от **теория**, с която да предадат необходимите знания за писане и четене на програмен код, и **практика** – разнообразни задачи, подбрани да затвърдят знанията и да формират в читателя навика, че винаги, когато пишем програми, мислим не само за синтаксиса, който ще използваме, а и за ефективното решение на проблема.

Езикът C# е подходящ избор, защото е един елегантен език, с който не се тревожим за представянето на нашата програма в паметта на компютъра, а можем да се концентрираме да подобряваме **ефективността** и елегант­ността на нашата програма.

Досега не съм попадал на книга за програмиране, която едновременно да запознава читателя с **езика** и да формира уменията му за **решаване на задачи**. Радвам се, че сега има такава книга, и съм сигурен че ще бъде изключително полезна на бъдещите програмисти.

***Васил Бакалов*** *е софтуерен инженер в* ***Microsoft Corporation****, Redmond, участник в проекта за първата българска книга за .NET: "Програмиране за .NET Framework".* Неговият блог е достъпен от <http://www.bakalov.com>. Можете да го откриете и в LinkedIn: [www.linkedin.com/in/vassilbakalov](https://www.linkedin.com/in/vassilbakalov).

### Отзив от Веселин Райчев, Google

Може би и без да прочетете тази книга ще можете да работите като софтуерен разработчик, но смятам, че ще ви е много по-трудно.

Наблюдавал съм случаи на **преоткриване на колелото**, много често в по-лош вид от теоретично най-доброто и най-често целият екип губи от това. Всеки, занимаващ се с програмиране, рано или късно трябва да прочете какво е **сложност на алгоритъм**, какво е **хеш-таблица**, какво **е двоично търсене** или практиките за използване на шаблони за проекти­ране (**design patterns**). Защо не започнете още отсега като прочетете тази книга?

Съществуват много книги за C# и още повече за програмиране. За много от тях ще кажат, че са най-доброто ръководство, най-бързо навлизане в езика. Тази книга е различна с това, че ще ви покаже **какво трябва да знаете**, за да постигате успехи, а не какви са тънкостите на даден език за програ­миране. Ако смятате темите в тази книга за безинтересни, вероятно софтуерното инженерство просто не е за вас.

***Веселин Райчев*** *е софтуерен инженер в* ***Google****, където се занимава с Google Maps и Google Translate. Преди това е работил в Motorola Biometrics и Metalife AG.*

*Веселин е печелил призови отличия в редица национални и* ***междуна­родни състезания*** *и е носител на бронзов медал от Международната олим­пиада по информатика, Южна Корея, 2002 и сребърен медал от Балкани­ада по информатика. Два пъти е представял СУ "Св. Климент Охридски" на световни финали по информатика (ACM ICPC) и е* ***преподавал*** *в няколко изборни курса във Факултета по математика и информатика на СУ. Можете да го откриете в LinkedIn:* [*https://www.linkedin.com/in/veselinr*](https://www.linkedin.com/in/veselinr)*.*

### Отзив от Васил Поповски, VMware

Като служител с ръководна роля във фирма **VMware** и преди това в Sciant често ми се налага да правя технически интервюта на кандидати за работа в нашата фирма. Учудващо е колко голяма част от кандидатите за софтуерни инженери, които идват на интервюта при нас, не владеят **фундаментални основи на програмирането**. Случва се кандидати с дълго­годишен опит да не могат да нарисуват **свързан списък**, да не знаят как работи **хеш-таблицата**, да не са чували какво е **сложност на алго­ритъм**, да не могат **да сортират масив** или да го сортират, но със сложност О(n3). Направо не е за вярване колко много самоуки програмисти има, които не владеят фундаментал­ните основи на прог­рамирането, които ще намерите в тази книга. Много от практику­ващите професията софтуерен разработ­чик не са наясно дори с **най-основните структури от данни** в програмира­нето и не знаят как да обходят дърво с рекурсия. **За да не бъдете като тях, прочетете тази книга!** Тя е първото учебно пособие, от което трябва да започнете своето развитие като програмисти. Фундамен­талните позна­ния по струк­тури от данни, алгоритми и решаване на задачи, които ще намерите в тази книга, ще са ви необходими, за да изградите успешно кариерата си на софтуерен разработчик и разбира се, да бъдете **успешни по интервютата** за работа и след това на работното си място.

Ако започнете от правене на динамични уеб сайтове с бази от данни и AJAX, без да знаете какво е свързан списък, дърво или хеш-таблица, един ден ще разберете какви **фундамен­тални пропуски** в знанията си имате. Трябва ли да се изложите на интервю за работа, пред колегите си или пред началника си, когато се разбере, че не знаете **за какво служи хеш-кодът** или **как работи структурата** List<Т> или как се обхождат рекур­сивно директориите по твърдия диск?

Повечето книги за програмиране ще ви научат да пишете прости програмки, но няма да обърнат внимание на **качеството на програмния код**. Това е една тема, която повечето автори смятат за маловажна, но писането на качествен код е основно умение, което **отличава кадърните от посред­ствените програмисти**. С годините можете и сами да стигнете до добрите практики, които тази книга ще ви препоръча, но трябва ли да се учите по метода на **пробите и грешките**? Тази книга ще ви даде лесния начин да тръгнете в правилната посока – да овладеете базовите структури от данни и алгоритми, да се научите да мислите правилно и да пишете кода си качествено. Пожелавам ви **ползотворно четене**.

***Васил Поповски*** *е софтуерен архитект във* ***VMware България*** *с повече от 10 години професионален опит като Java разработчик. Във VMware България се занимава с разработка на скалируеми, Enterprise Java системи. Преди това е работил като старши мениджър във VMware България, като технически директор във фирма* ***Sciant*** *и като ръководител екип в* ***SAP Labs България****.*

*Като ученик Васил е печелил призови отличия в редица национални и* ***международни състезания*** *и е носител на бронзов медал от Междуна­родната олимпиада по информатика, Сетубал, 1998 и бронзов медал от Балканиада по информатика, Драма, 1997. Като студент Васил участвал в редица национални студентски* ***състезания*** *и в световното междууниверси­тетско състезание по програмиране (ACM ICPC). През 2001/2002 води курса "Обработване на транзакции" в СУ "Св. Климент Охридски". Васил е един от учредителите на Българска асоциация на разработчиците на софтуер (БАРС).*

### Отзив от Павлин Добрев, ProSyst Labs

Книгата "Въведение в програмирането със C#" е отлично **учебно пособие за начинаещи**, което ви дава възможност по лесен и достъпен начин да овладеете осно­вите на програмирането. Това е шестата книга, написана под ръковод­ството на Светлин Наков, и също както останалите, е изключително ориентирана към усвояването на **практически умения** за програмиране. Учебното съдържание обхваща фундаментални теми като структури от данни, алгоритми и решаване на задачи и това я прави непреходна при развитието на технологиите. Тя е изпълнена **с много­бройни примери** и **практически съвети** за решаване на основни задачи от ежедневната работа на един програмист.

Книгата "**Въведение в програмирането със C#**" представлява адаптация към езика C# и платформата Microsoft .NET Framework на изключително успеш­ната книга "Въведение в програ­мирането с Java" и се базира на натрупания опит на водещия автор **Светлин Наков** в преподаването на основи на програмирането – както в Националната академия по разработка на софтуер (НАРС) и по-късно в Telerik Academy и [СофтУни](http://softuni.bg), така и във ФМИ на Софийски университет "Св. Климент Охридски", Нов български универ­ситет (НБУ) и Технически университет-София.

Въпреки големия брой автори, всеки от които с различен професионален и преподавателски опит, между отделните глави на книгата се забелязва ясна логическа свързаност. Тя е написана **разбираемо**, с подробни обяснения и с **много, много примери**, далеч от сухия академичен стил, присъщ за повечето университетски учебници.

Насочена към прохождащите в програмирането, книгата поднася внима­телно, **стъпка по стъпка**, най-важното, което един програмист трябва да владее, за да практикува професията си – започвайки от променливи, цикли и масиви и достигайки до **фундаменталните структури от данни** и **алгоритми**. Книгата засяга и важни теми като рекурсивни алгоритми, дървета, графи и хеш-таблици. Това е една от малкото книги, която същевременно учи на добър програмен стил и **качествен програмен код**. Отделено е достатъчно внимание на принципите на обектно-ориенти­раното програмиране и **обработката на изключения**, без които съвремен­ната разработка на софтуер е немислима.

Книгата "Въведение в програмирането със C#" учи на важните **принципи и концепции в програмирането**, на начина, по който програмистите разсъж­дават логически, за да решават проблемите, с които се сблъскват в еже­дневната си работа. Ако трябваше заглавието на книгата да съответства още по-точно на съдържанието й, тя трябваше да се казва "**Фундамен­тални основи на програмирането**".

Тази книга не съдържа всичко за програмирането и няма да ви направи .NET софтуерни инженери. За да станете **наистина добри програмисти**, ви трябва **много, много практика**. Започнете от задачите за упражнения след всяка глава, но не се ограничавайте само с тях. Ще изпишете хиляди редове програ­мен код докато наистина станете добри – такъв е животът на програмиста. Тази книга е наистина **силен старт**! Възползвайте се от възможността да намерите всичко най-важно на куп, без да се лутате из хилядите самоучи­тели и статии в Интернет. На добър път!

***Д-р Павлин Добрев*** *е технически директор на фирма* ***Просист Лабс*** *(*[*www.prosyst.com*](http://www.prosyst.com)*), придобита през 2015 г. от немския гигант* [***Bosch***](http://www.bosch.com/)*, софтуерен инженер с повече от 15 години опит, консултант и учен, доктор по Компютърни системи, комплекси и мрежи. Павлин има световен принос в развитието на съвременните* ***компютърни технологии*** *и* ***технологични стандарти****. Той участва активно в междуна­родни стандартизационни организации като OSGi Alliance (*[*www.osgi.org*](http://www.osgi.org)*) и Java Community Process (*[*www.jcp.org*](http://www.jcp.org)*), както и инициативи за софтуер с отворен код като Eclipse Foundation (*[*www.eclipse.org*](http://www.eclipse.org)*). Павлин управлява софтуерни проекти и консултира фирми като Miele, Philips, Siemens, BMW, Bosch, Cisco Systems, France Telecom, Renault, Telefonica, Telekom Austria, Toshiba, HP, Motorola, Ford, SAP и др. в областта на вградени приложения, OSGi базирани системи за автомобили, мобилни устройства и домашни мрежи, среди за разработка и Java Enterprise сървъри за приложения. Той има* ***докторска степен*** *(PhD) по компютърни науки, автор е на много научни и технически публикации и е участник в престижни международни конференции.*

### Отзив от Николай Манчев, Oracle

За да станете **добър разработчик на софтуер**, трябва да имате готовност да инвестирате в натрупването на познания в няколко области и кон­кретния език за програмиране е само една от тях. Добрият разработчик трябва да познава не само синтаксиса и приложно-програмния интерфейс на езика, който си е избрал. Той трябва да притежава също така задълбочени познания по **обектно-ориентирано програмиране**, **струк­тури от данни** и **писане на качествен код**. Той трябва да подкрепи тези си познания и със **сериозен практически опит**.

Когато започвах своята кариера на разработчик на софтуер преди повече от 15 години, намирането на **цялостен източник**, от който да науча тези неща беше **невъзможно**. Да, тогава имаше книги за отделните програмни езици, но те описваха единствено техния синтаксис. За описание на приложно-програмния интерфейс трябваше да се ползва самата докумен­тация към библиотеките. Имаше отделни книги посветени единствено на обектно-ориентираното програмиране. Различни алгоритми и структури от данни пък се преподаваха в университета. За качествен програмен код не се говореше въобще.

Научаването на всички тези неща "на парче" и усилията по събирането им в единен контекст си оставаше работа на избралия "пътя на програмиста". Понякога един такъв самообразоващ се програмист не успява да запълни огромни пропуски в познанията си, просто защото няма идея за тяхното съществуване. Нека ви дам един пример, за да илюстрирам проблема.

През 2000 г. поех да управлявам един **голям Java проект**. Екипът, който го разработваше беше от 25 души и до момента по проекта имаше написани приблизително 4 000 Java класа. Като ръководител на екипа, част от моята работа включваше редовното преглеждане на кода написан от другите програмисти. Един ден видях как един от моите колеги беше решил стандартната задача по **сортиране на масив**. Той беше написал отделен метод от около 25 реда, който реализираше тривиалния алгори­тъм за сортиране по метода на мехурчето. Когато отидох при него и го запитах защо е направил това вместо да реши проблема на един един­ствен ред използвайки Arrays.sort(), той се впусна в обяснения как вградения метод е по-тромав и е по-добре тези неща да си ги пишеш сам. Накарах го да отвори документацията и му показах, че "тромавият" метод работи със **сложност O(n\*log(n))**, а неговото мехурче е еталон за лоша производи­телност със своята **сложност O(n\*n)**. В следващите няколко минути от нашия разговор направих и истинското откритие – моят колега нямаше идея какво е **сложност на алгоритъма**, а самите му познания по стандартни алгоритми бяха трагични. В последствие открих, че той е завършил съвсем друг тип инженерна специалност, а не информатика. В това, разбира се, няма абсолютно нищо лошо. В познанията си по Java той не отстъпваше на останалите колеги, които имаха по-дълъг практически опит от него. Но в този ден ние открихме празнина в неговата квалифи­кация на разработчик, за която той не беше и подозирал.

Не искам да оставате с погрешни впечатления от тази история. Въпреки че един студент издържал успешно основните си изпити по специалност "Информатика" в добър университет със сигурност ще знае базовите алгоритми за сортиране и ще може да изчисли тяхната сложност, той също ще има своите пропуски. Тъжната истина е, че в България универси­тетското образование по тази специалност все още е с твърде **теоретична насоченост**. То твърде малко се е променило за последните 15 години. Да, програмите вече се пишат на Java и C#, но това са същите програми, които се пишеха тогава на Pascal и Ada.

Преди около година приех за консултация студент първокурсник, който следваше в специалност "Информатика" на един от най-големите дър­жавни университети в България. Когато седнахме да прегледаме заедно записките му от лекциите по "**Увод в програмирането**" бях изумен от примерния код даван от преподавателя. Имената на методите бяха смесица от английски и транслитериран български. Имаше метод calculate и метод rezultat. Променливите носеха описателните имена a1, a2, и suma. Да, в този подход няма нищо трагично, докато се използва за примери от десет реда, но когато този студент заеме след години своето заслужено място в някой голям проект, той ще бъде тежко порицан от ръководителя на проекта, който ще му обяснява за **код конвенции**, **именуване на променливи**, методи и класове, **логическа свързаност на отговорностите** и **диа­пазон на активност**. Тогава те заедно ще открият неговата празнина в познанията по качествен код по същия начин, по който ние с моя колега открихме проблемните му познания в областта на алгоритмите.

Скъпи читателю, смело мога да заявя, че **в ръцете си държиш една наистина уникална книга**. Нейното съдържание е подбрано изключи­телно внимателно. То е подредено и поднесено с внимание към детайлите, на който са способни само хора с огромен практически опит и солидни научни познания като водещите автори на тази книга **Светлин Наков** и **Веселин Колев**. Години наред те също са се учили "в движение", допълвайки и разширявайки своите познания. Работили са години по огромни софтуерни проекти, участвали са в научни конференции**, препо­да­вали са на стотици студенти**. Те знаят какво е нужно да знае всеки един, който се стреми към кариера в областта на разработката на софтуер и са го поднесли така, както никоя книга по увод в програмирането не го е правила до момента. Твоето пътуване през страниците ще те преведе през синтаксиса на **езика C#**. Ще видиш използването на голяма част от приложно-програмния му интерфейс (API). Ще научиш основите на обектно-ориентираното програмиране и ще боравиш свободно с термини като **обекти**, **събития** и **изключения**. Ще видиш най-често използваните **структури от данни** като **масиви**, **дървета**, **хеш-таблици** и **графи**. Ще се запознаеш с най-често използваните **алгоритми** за работа с тези структури и ще узнаеш за техните плюсове и минуси. Ще разбереш концепциите по конструиране на **качествен програмен код** и ще знаеш какво да изискваш от програмистите си, когато някой ден станеш ръководител на екип. В допълнение книгата ще те предизвика с много **практически задачи**, които ще ти помогнат да усвоиш по-добре и по пътя на практиката материала, който се разглежда в нея. А ако някоя от задачите те затрудни, винаги ще можеш да погледнеш решението, което авторите предоставят за всяка от тях.

Програмистите правят грешки – от това никой не е застрахован. По-добрите грешат от недоглеждане или преумора, a по-лошите – от незнание. Дали ще станеш **добър или лош разработчик** на софтуер зависи изцяло от теб и най-вече от това, доколко си готов постоянно да инвес­тираш в своите познания – било чрез курсове, чрез четене или чрез практическа работа. Със сигурност обаче мога да ти кажа едно – колкото и време да инвестираш в тази книга, няма да сгрешиш. Ако преди няколко години някой, желаещ да стане разработчик на софтуер, ме попиташе "От къде да започна?", нямаше как да му дам еднозначен отговор. Днес мога без притеснения да заявя – "**Започни от тази книга!** (във варианта й за C# или Java)".

От все сърце ти желая успех в овладяването на тайните на C#, .NET Framework и разработката на софтуер!

***Николай Манчев*** *е консултант и софтуерен разработчик с дългогодишен опит в Java Enterprise и Service Oriented Architecture (SOA). Работил е за* ***BEA Systems*** *и* ***Oracle Corporation****. Той е сертифициран разработчик по програмите на Sun, BEA и Oracle. Преподава софтуерни технологии и води курсове по Мрежово програмиране, J2EE, Компресия на данни и Качествен програмен код в ПУ "Паисий Хилендарски" и СУ "Св. Климент Охридски". Водил е редица курсове за разработчици по Oracle технологии в централна и източна Европа (Унгария, Гърция, Словакия, Словения, Хърватска и други) и е участвал в международни проекти по внедряване на J2EE базирани системи за управление на сигурността. Негови раз­работки в областта на алгоритмите за компресия на данни са приети и представяни в САЩ от IEEE. Николай е почетен член на Българска асоциация на разработчиците на софтуер (БАРС). Автор е на книгата "Сигурност в Oracle Database : Версия 10g и 11g". Повече за него можете да намерите на личния му уеб сайт:* [*http://www.manchev.org*](http://www.manchev.org/)*. За да се свържете с него използвайте e-mail:* [*nick@manchev.org*](mailto:nick@manchev.org)*.*

### Отзив от Панайот Добриков, SAP AG

Настоящата книга е едно изключително добро въведение в програми­рането за начинаещи и водещ пример в течението (промоцирано от Wikipedia и други) да се създава и разпространява **достъпно за всеки знание** не само \***безплатно**\*, но и с **изключително високо качество**.

***Панайот Добриков*** *е програмен директор в* ***SAP AG*** *и съавтор на книгата "Програмиране=++Алгоритми;". Повече за него можете да намерите на личния му уеб сайт:* [*http://indyana.hit.bg*](http://indyana.hit.bg/)*.*

### Отзив от Любомир Иванов, Mobiltel

Ако преди 5 или 10 години някой ми беше казал, че съществува книга, от която да научим основите на управлението на хора и проекти – бюджетиране, финанси, психология, планиране и т.н., нямаше да му повярвам. Не бих повярвал и днес. За всяка от тези теми има десетки книги, които трябва да бъдат прочетени.

Ако преди година някой ми беше казал, че съществува книга, от която можем да научим **основите на програмирането**, необходими на всеки софтуерен разработчик, нямаше да му повярвам.

Спомням си времето като начинаещ програмист и студент – четях няколко книги за езици за програмиране, други за алгоритми и структури от данни, а трети за писане на качествен код. Много малко от тях ми помог­наха **да мисля алгоритмично** и да си изградя подход за решаване на ежедневните проблеми, с които се сблъсквах в практиката. Нито една не ми даде **цялостен поглед над всичко**, което исках и трябваше да знам като програмист и софтуерен инженер. Единственото, което помагаше, беше инатът и преоткриването на колелото.

Днес чета тази книга и се радвам, че най-сетне, макар и малко късно за мен, **някой се е хванал и е написал Книгата**, която ще помогне на всеки начинаещ програмист да сглоби **големия пъзел на програмирането** – моде­рен език за програмиране, структури от данни, качествен код, алгорит­мично мислене и решаване на проблеми. Това е **книгата, от която трябва да започнете с програмирането**, ако искате да овладеете изкуството на качественото програмиране. Дали ще изберете C# или Java варианта на тази книга няма особено значение. Важното е да се научите **да мислите като програмисти** и да решавате проблемите, които възникват при писа­нето на софтуер, а езикът е само един инструмент, който можете да смените с друг по всяко време.

Тази книга **не е само за начинаещите**. Дори програмисти с няколкогоди­шен опит има какво да научат от нея. Препоръчвам я на всеки разработ­чик на софтуер, който би искал да разбере какво не е знаел досега.

Приятно четене!

***Любомир Иванов*** *е ръководител на отдел "Data and Mobile Applications" в* ***Мобилтел ЕАД****, където се занимава с разработка и внедряване на ИТ решения за telecom индустрията.*

### Отзив от Христо Дешев

Учудващо е, че голям процент от програмистите не обръщат внимание на малките неща като **имената на променливите** и добр**ата структура на кода**. Тези неща се натрупват и накрая формират разликата между добре напи­сания софтуер и купчината спагети. Тази книга учи на **дисциплина** и "**хигиена**" в писането на код още с основите на програмирането, а това несъмнено ще Ви изгради като **професионалист**.

***Христо Дешев****, software craftsman*

## Принос на Telerik: A Progress Company

Значим принос към реализирането на книгата "Въведение в програми­рането със C#" има водещата софтуерна компания [Telerik: A Progress Company](http://www.telerik.com). **Ръководството на** **Telerik подкрепя книгата** още когато тя съществува само като идея и прави издаването й възможно благодарение на финансовата и моралната помощ, които оказва. Голяма част от авторите на "Въведение в програмирането със C#" са служители на Telerik, които допринасят безвъзмездно с труда си, дори в работно време. В постоянното актуализиране на съдържанието на книгата и в нейния превод на английски език са участвали и курсисти от ["Академията на Телерик"](http://academy.telerik.com), образователна инициатива на Telerik за софтуерни инженери, които в рамките на обучението си помагат за популяризиране на проекта.

## Лиценз

Книгата и учебните материали към нея се разпространяват свободно по следния лиценз:

### Общи дефиниции

1. Настоящият лиценз дефинира **условията за използване** и разпрост­ранение на учебни материали и книга "Въведение в програмирането със C#", разработени от екип под ръководството на Светлин Наков ([www.nakov.com](http://www.nakov.com)) и Веселин Колев (<http://veskokolev.com>).
2. **Учебните материали** се състоят от:

* книга (учебник) по "Въведение в програмирането със C#";
* примерен сорс-код;
* демонстрационни програми;
* задачи за упражнения.

1. Учебните материали са достъпни за **свободно** изтегляне при усло­вията на настоящия лиценз от официалния сайт на проекта:

<http://www.introprogramming.info>

1. **Автори** на учебните материали са лицата, взели участие в тяхното изработване.
2. **Потребител** на учебните материали е всеки, който по някакъв начин използва тези материали или части от тях.

### Права и ограничения на потребителите

1. Потребителите **имат** право:

* да **разпространяват** безплатно непроменени копия на учебните материали в електронен или хартиен вид;
* да **използват** учебните материали или части от тях, включително примерите и демонстрациите, включени към учебните материали или техни модификации, за всякакви нужди, включително и в комерсиални проекти, при условие че **посочват оригиналния източник**, оригиналния автор на съответния текст или програмен код, настоящия лиценз и сайта [www.introprogramming.info](http://www.introprogramming.info);
* да **разпространяват** безплатно извадки от учебните материали или техни модифицирани копия (включително да ги превеждат на чужди езици или да ги адаптират към други програмни езици и платформи), но само при изричното **споменаване на оригинал­ния първоизточник** и авторите на съответния текст, програмен код или друг материал, настоящия лиценз и официалния сайт на проекта – [www.introprogramming.info](http://www.introprogramming.info).

1. Потребителите **нямат** право:

* да разпространяват срещу **заплащане** учебните материали или части от тях, като изклю­чение прави само програмният код;
* да **премахват настоящия лиценз** от учебните материали, когато ги модифицират за свои нужди.

### Права и ограничения на авторите

1. Всеки автор притежава **неизключителни права** върху продуктите на своя труд, с които взима участие в изработката на учебните мате­риали.
2. Авторите имат право **да използват частите, изработени от тях**, за всякакви цели, включително да ги изменят и разпространяват срещу заплащане.
3. Правата върху учебните материали, изработени в **съавторство**, са притежание на всички съавтори заедно.
4. Авторите **нямат право да разпространяват срещу заплащане** учебни материали или части от тях, изработени в съавторство, без изрич­ното съгласие на всички съавтори.

## Сайтът на книгата

Официалният уеб сайт на книгата "Въведение в програмирането със C#" е достъпен от адрес: <http://www.introprogramming.info>. От него можете да изтеглите цялата книга в електронен вид, сорс кода на примерите и други полезни ресурси.

## Форум за въпроси по книгата

Форумът за **въпроси по книгата** е достъпен от <http://softuni.bg/forum>. В него ще намерите решение на почти всички задачи от книгата. Този форум е създаден за дискусия между участниците в курсовете от [СофтУни](http://softuni.bg), които в първите няколко месеца на своето обучение преминават през почти целия учебен материал от настоящата книга и решават задачите от упражненията.

Във форума ще намерите както **коментари и решения**, изпратени от студенти и читатели на книгата, така и от авторитетни преподаватели от [СофтУни](http://softuni.bg). Просто се разровете дос­татъчно задълбочено в архивите на форума и ще намерите по няколко **решения на всички задачи** от книгата (които са адаптирани в курсовете на [Софтуерния университет](http://softuni.bg)). Всяка година няколко хиляди участници в курсовете на СофтУни решават задачите от тази книга и споделят решенията и трудностите, с които са се сблъскали във форума, така че просто търсете усърдно в архивите, ако не можете да се справите с някоя задача. Ако не намерите нищо, **питайте** и ще ви отговорят. Колегите са много дружелюбни и помагат постоянно.

## Видеоматериали за самообучение по книгата

Книгите са прекрасен учебен ресурс. Още по-прекрасен учебен ресурс за обаче **видео-уроците**. Те представят учебния материал нагледно, показват ви как се правят нещата на живо, демонстрират ви кое точно как се случва. Затова ви препоръчваме да добавите към обучението си по тази книга и гледането на **видео уроци от Софтуерния университет** и по-конкретно следните курсове:

* **Основи на програмирането**:  
  [https:softuni.bg/courses/programming-basics/](https://softuni.bg/courses/programming-basics/)
* **Обектно-ориентирано програмиране**:  
  <https://softuni.bg/courses/oop/>
* **Качествен програмен код**:  
  <https://softuni.bg/courses/high-quality-code/>

## Фен клуб на книгата

Фен клубът на книгата е нейната **страница във Facebook**, достъпна от <https://www.facebook.com/IntroCSharpBook>. Там се е формирала онлайн общност от бъдещи и настоящи програмисти, които харесват "Въведение в програмирането със C#" и с които можете да обсъждате учебния материал.

Светлин Наков,

Мениджър "обучение и вдъхновение"

Софтуерен университет (СофтУни),

11.04.2015 г.

# 