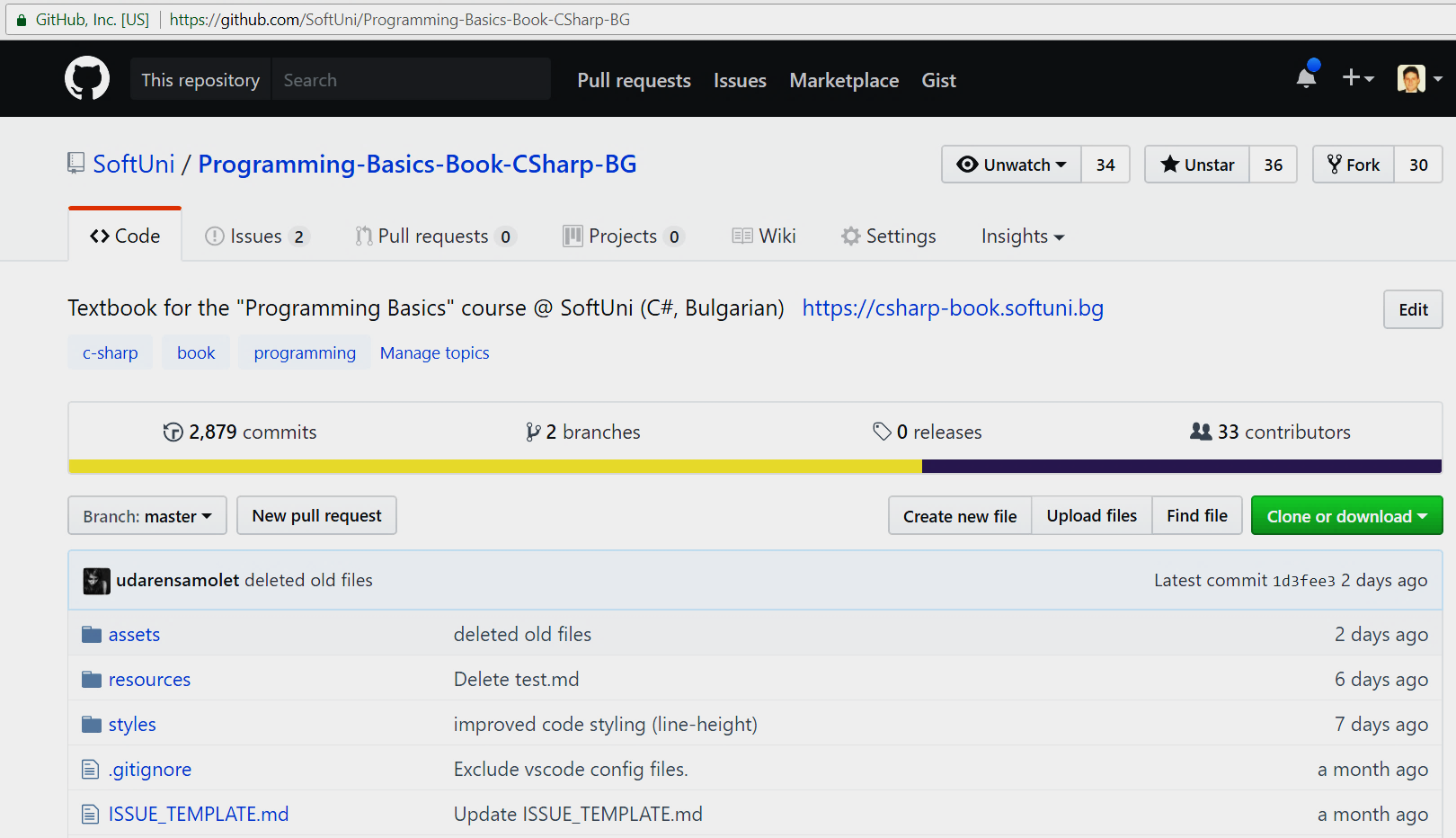
# Подготовка на книгата за печат

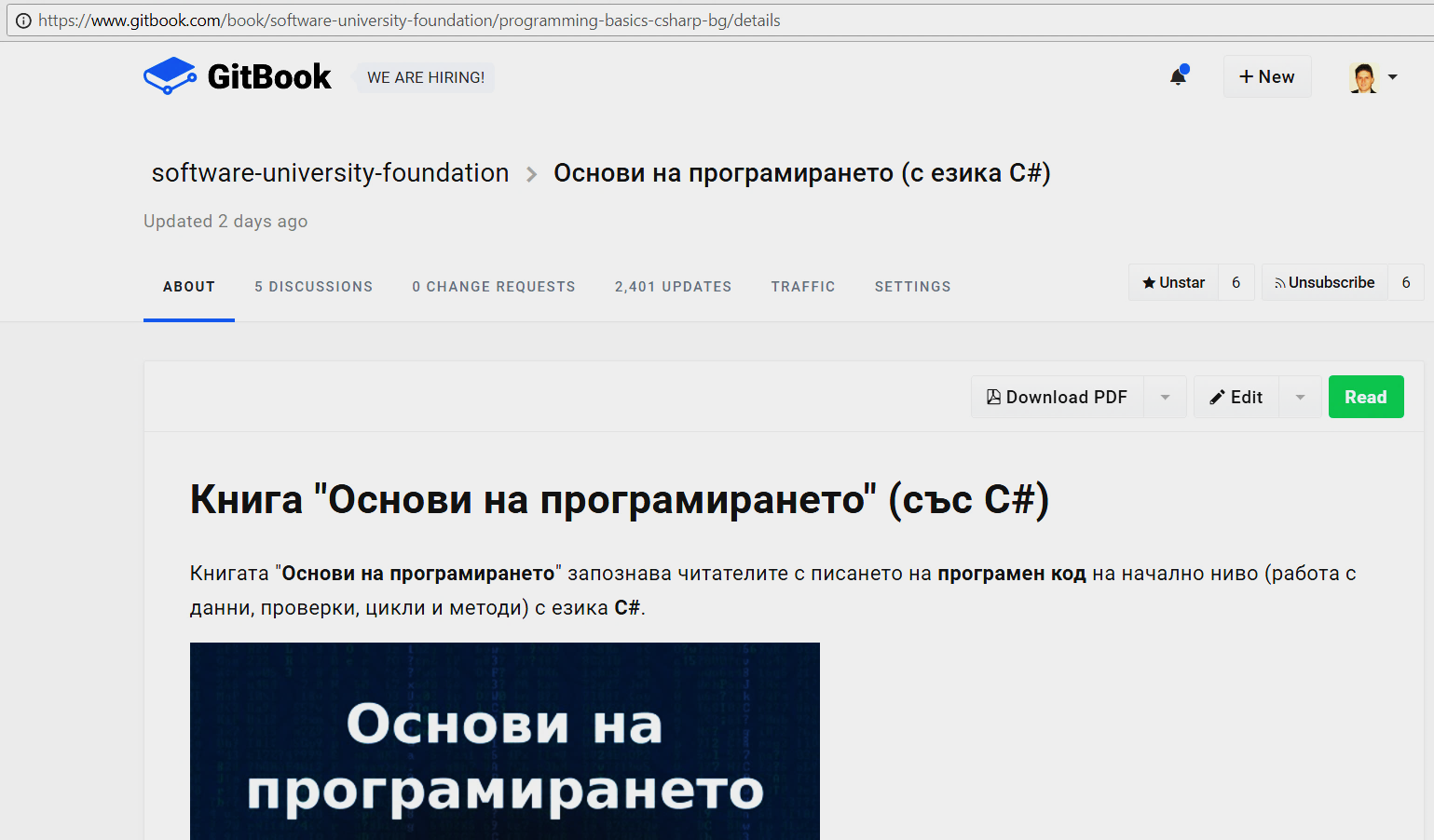
Настоящото ръководство обяснява процеса на подготовка на “Programming Basics” книгата за печат: конвертиране от GitHub **Markdown** формат към **HTML**, след това към **MS Word DOCX** и корекции и **оформяне на DOCX файла**.

## Конвертиране от Markdown към HTML

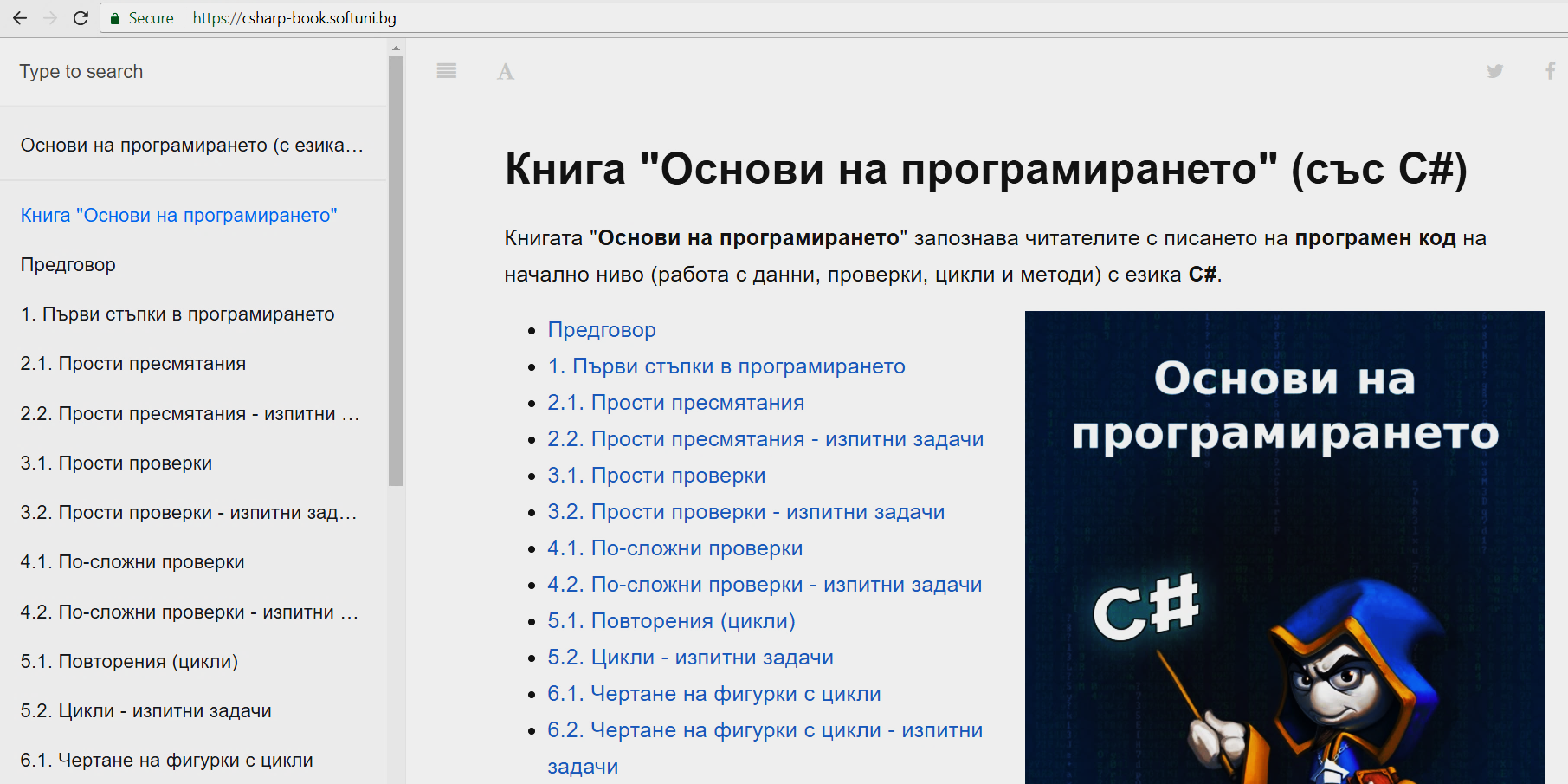
Книгата се разработва в **Markdown** формат като публичен проект в GitHub: <https://github.com/SoftUni/Programming-Basics-Book-CSharp-BG>.



**Версията за четене** се генерират от инфраструктурата на GitBook, чрез предварително конфигуриран GitBook проект: <https://www.gitbook.com/book/software-university-foundation/programming-basics-csharp-bg/details>.



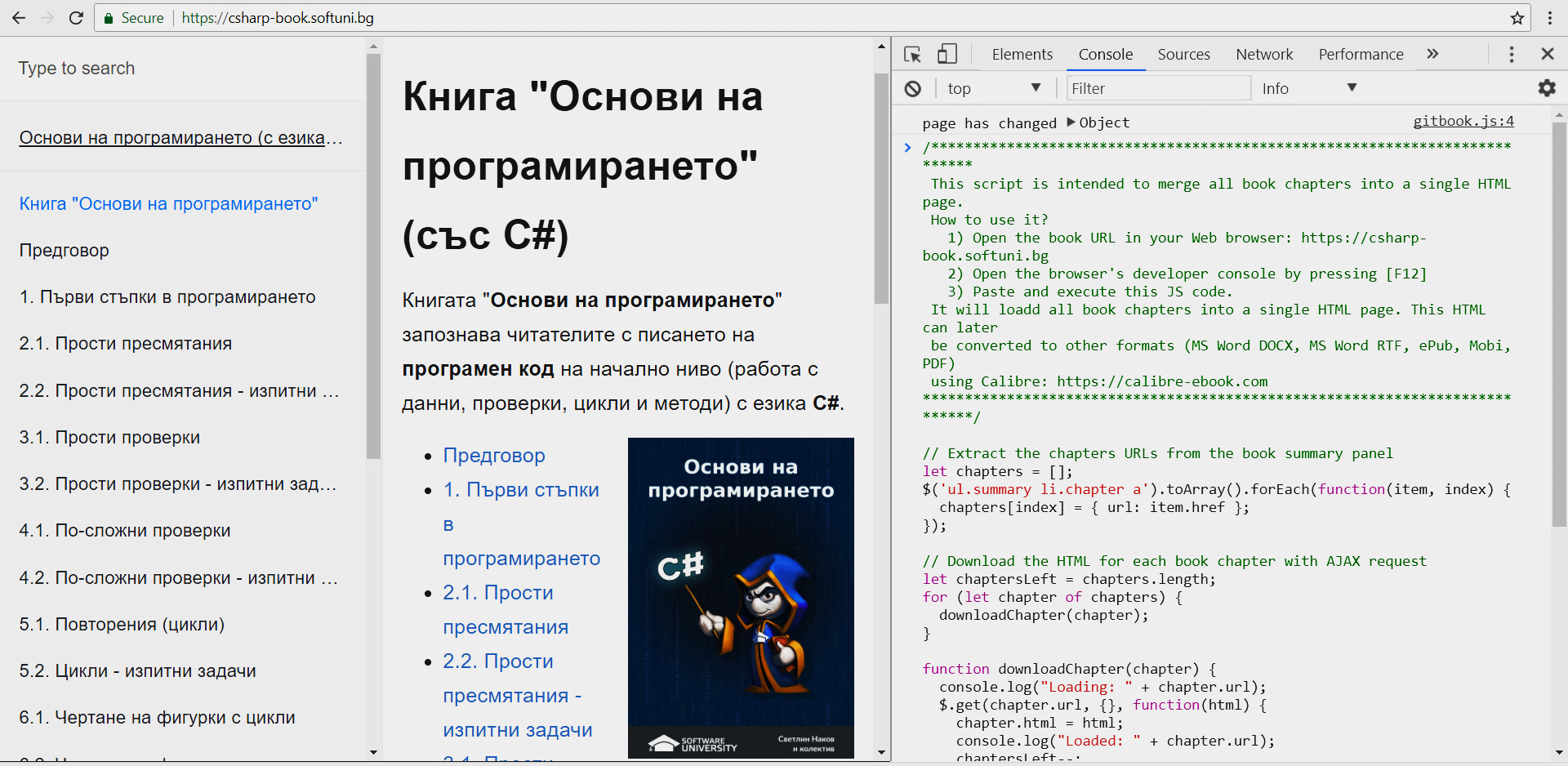
GitBook произвежда **HTML книга** с добро качество, разделена на глави: <https://csharp-book.softuni.bg>.

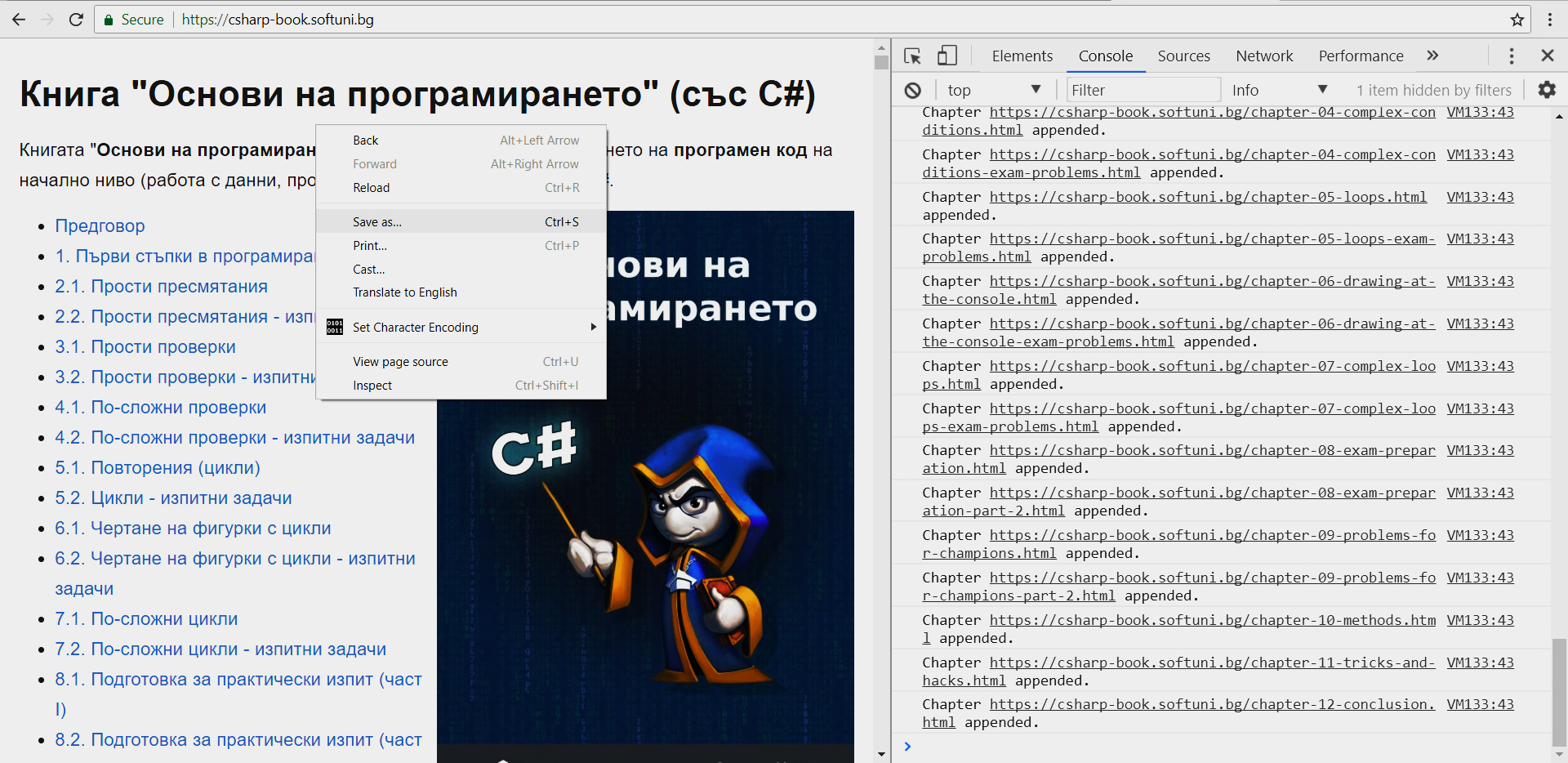


## Съединяване на всички глави в общ HTML файл

За съединяването на всички глави на книгата в общ HTML файл, подходящ за печат или последваща обработка се използва **client-side JavaScript код**: <https://github.com/SoftUni/Programming-Basics-Book-CSharp-BG/blob/master/resources/Prepress-Scripts/merge-all-book-chapters.js>. Кодът се изпълнява по следния начин:

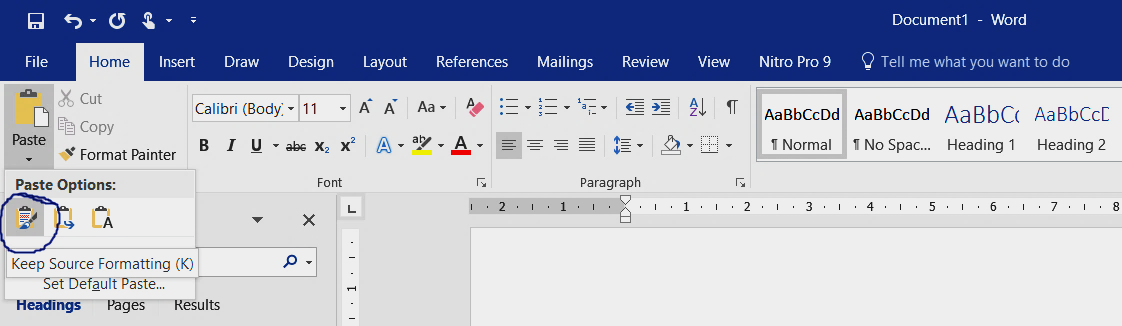
1. **Отваря се книгата** в HTML вариант от нейния сайт: <https://csharp-book.softuni.bg>.
2. **Отваря се конзолата** на уеб браузъра (например Google Chrome Console се отваря с натискане на [F12]). Избира се секцията [**Console**].
3. В конзолата се **изпълнява скрипта** [merge-all-book-chapters.js](https://github.com/SoftUni/Programming-Basics-Book-CSharp-BG/blob/master/resources/Prepress-Scripts/merge-all-book-chapters.js).
4. Цялата книга се зарежда като **единична уеб страница** в браузъра (всички глави една след друга).
5. Резултатът се **записва като HTML файл** (чрез [Ctrl+S] в Google Chrome).



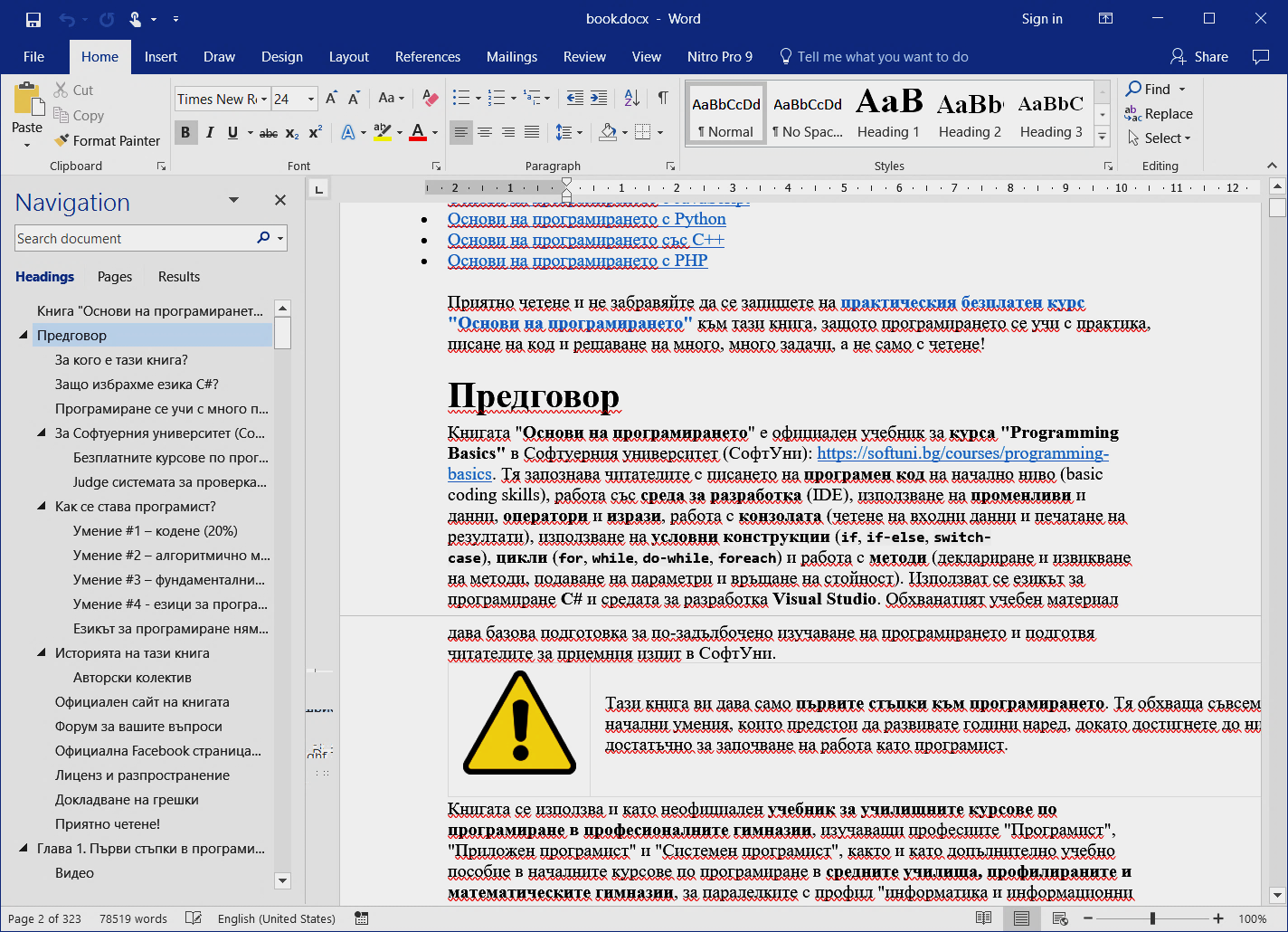


## Конвертиране от HTML към MS Word

* **Копираме HTML текста** от браузъра в **clipboard** (с натискане на [Ctrl+C] в Google Chrome).
* Създаваме **нов MS Word документ**, чрез копиране на шаблонния файл [book-template.docx](https://github.com/SoftUni/Programming-Basics-Book-CSharp-BG/blob/master/resources/Prepress-Scripts/book-template.docx).
* **Поставяме (paste)** копирания HTML сорс код на книгата от браузъра в MS Word, като **запазваме оригиналното форматиране**:



Получава се доста **счупен MS Word файл** с много проблеми по форматирането:



## Корекции по MS Word файла (автоматични)

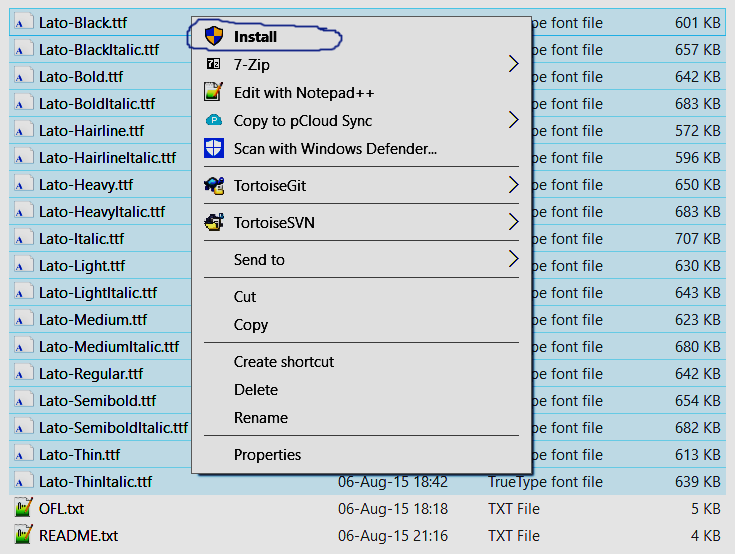
Експортът на книгата от HTML към MS Word има много **дефекти** и проблеми за **оправяне и подобряване** и затова е написана **C# програма за предпечат**.

### Инсталиране на шрифт Lato

Преди да се започне с предпечата на книгата, трябва **да се инсталират** Lato **шрифтовете**, които са достъпни от хранилището на проекта:

* <https://github.com/SoftUni/Programming-Basics-Book-CSharp-BG/blob/master/resources/Prepress-Scripts/Lato-font-family-TTF.zip>

Това са **стандартни TTF файлове**, които се инсталират в Windows с **[Install]** от Windows Explorer:

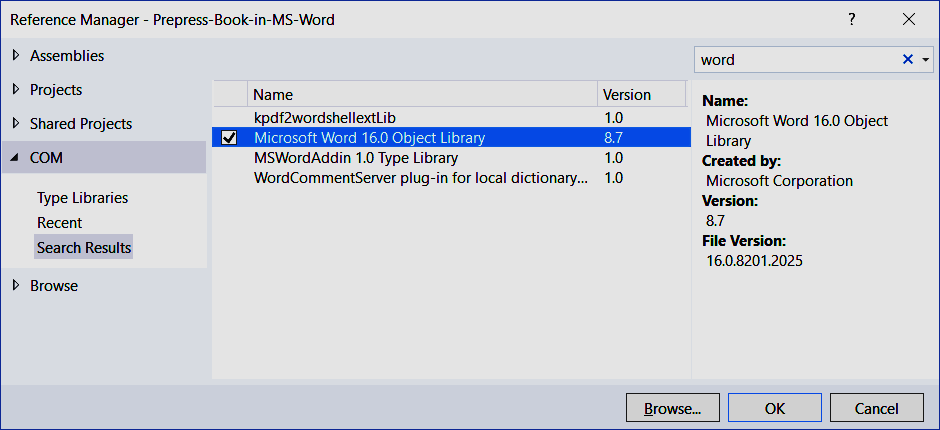


### C# програма за предпечат на книгата

Експортът на книгата от HTML към MS Word има доста **дефекти за оправяне** и подобряване и затова е написана **C# програма за предпечат**, която ползва **Word.Interop API**, за да управлява MS Word (през automation API) и да го кара да отвори документа и да направи серия промени в него, с които проблемите един по един се оправят. Ето тук е **сорс кодът** на C# програмата за предпечатно оформление:

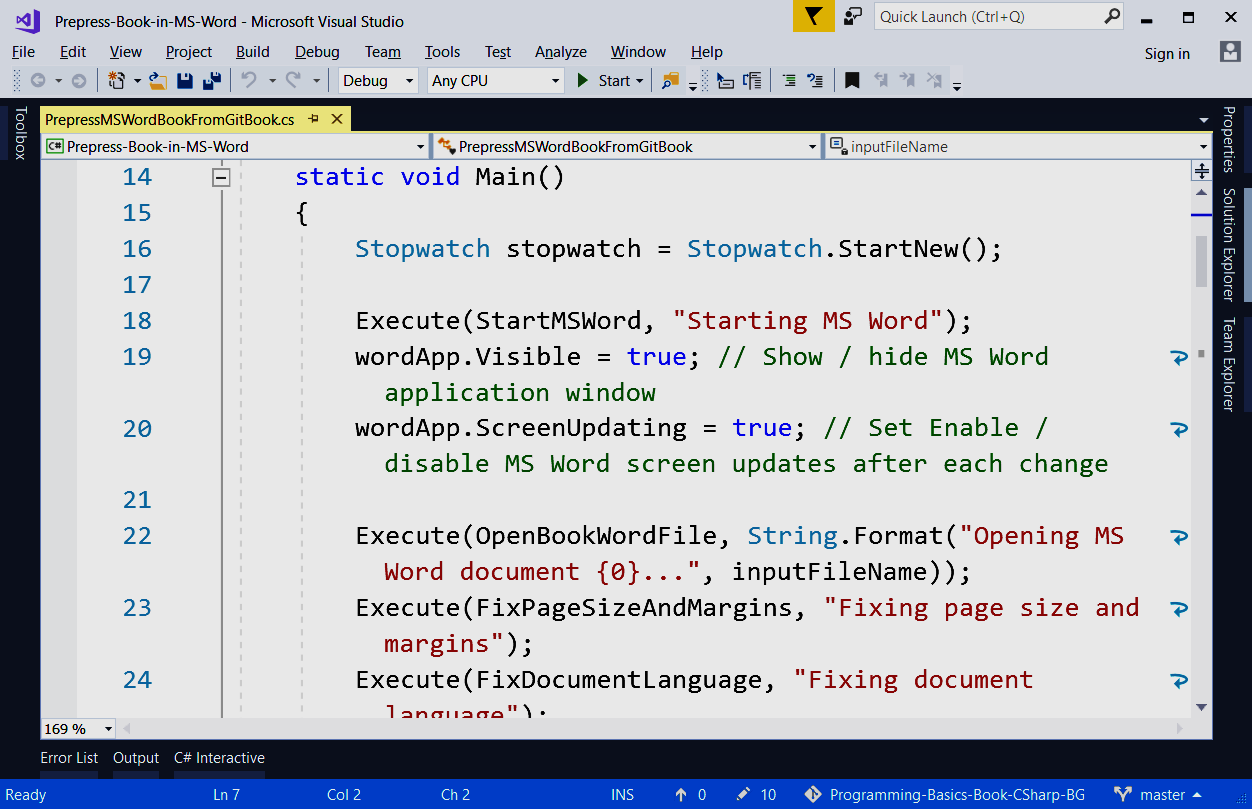
* <https://github.com/SoftUni/Programming-Basics-Book-CSharp-BG/tree/master/resources/Prepress-Scripts>

Проектът реферира ActiveX обекта (COM контролата) “**Microsoft Word 16.0 Object Library**”:



В нея се съдържа namespace Microsoft.Office.Interop.Word, в който са класовете и интерфейсите за работа с MS Word документи и обекти: Application, Document, Paragraph, InlineShape и други.

Когато бъде отворен **Visual Studio проектът**, той изглежда по следния начин:



### Какво оправя C# програмата за предпечат?

Програмата ползва **Word.Interop API** и обработва съществуващ **DOCX файл**. Тя оправя последователно поредица от дефекти, описани по-долу.

#### Размер на страницата и отстъпи

* Размер на страницата: **25 cm x 18 cm**
* Отстъпи: 1 cm (top, down), 1.5 cm (inner), 1 cm (outer)
* Header / Footer: 1 cm отстъп

#### Корекции в заглавията

* Всички заглавия се поправят да ползват стиловете **Heading 1**, **Heading 2**, …, **Heading 6**.
* Включва се ползване на шрифтове “**Lato**”.
* **Заглавия**: **Lato Medium** (без да се цъка bold)
* **Текст**: **Lato Light**, justified paragraph

#### Размери на картинките

* Поправят се размерите на картинките: 100% за по-малките и page width за по-големите.

#### Форматиране на таблиците

* Форматирате се коректно всички таблици, така че да не излизат от страницата и да изглеждат добре.

#### Корекции в списъците

* Всички списъци (номерирани и неномерирани) се коригират, така че да изглеждат добре.

#### Форматиране на блоковете сорс код

* Форматирате се коректно блоковете сорс код, така че да изглеждат добре.
* Сорс кодът се отбелязва да не се проверява за правопис.

#### Език на текста

* На всички думи се задава правилният език: за кирилица 🡪 български, за останалите 🡪 английски.
* Всички думи, които приличат на сорс-код идентификатори, се отбелязват да се не проверяват за правопис.

## Корекции по MS Word файла (ръчни)

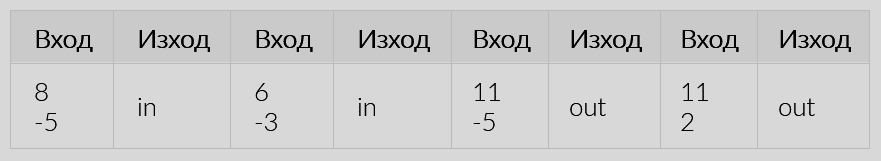
Не всичко по книгата може или има смисъл да се автоматизира програмно, така че някои **корекции** се правят **допълнително на ръка**. Ето и по-важните от тях.

#### Сгъстяване на таблиците с вход и изход

* Таблиците, съдържащи примерен вход и изход се **сгъстяват**, така че **да се използва минимално място от страницата** (да се пести хартия). Например следните две таблици със сорс код се обединяват:

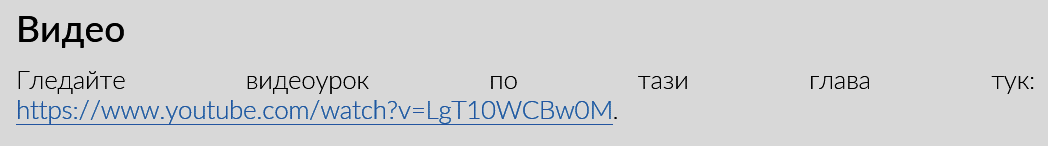


Резултатът е таблица, която заема **по-малко място**:



#### Оправят се празнините в текста

Когато има твърде много **празно място в текста**, той изглежда грозно. Пример:



Празнината може да де оправи по няколко начина. Може с **[Shift + Enter]** да се премина на **нов ред**:

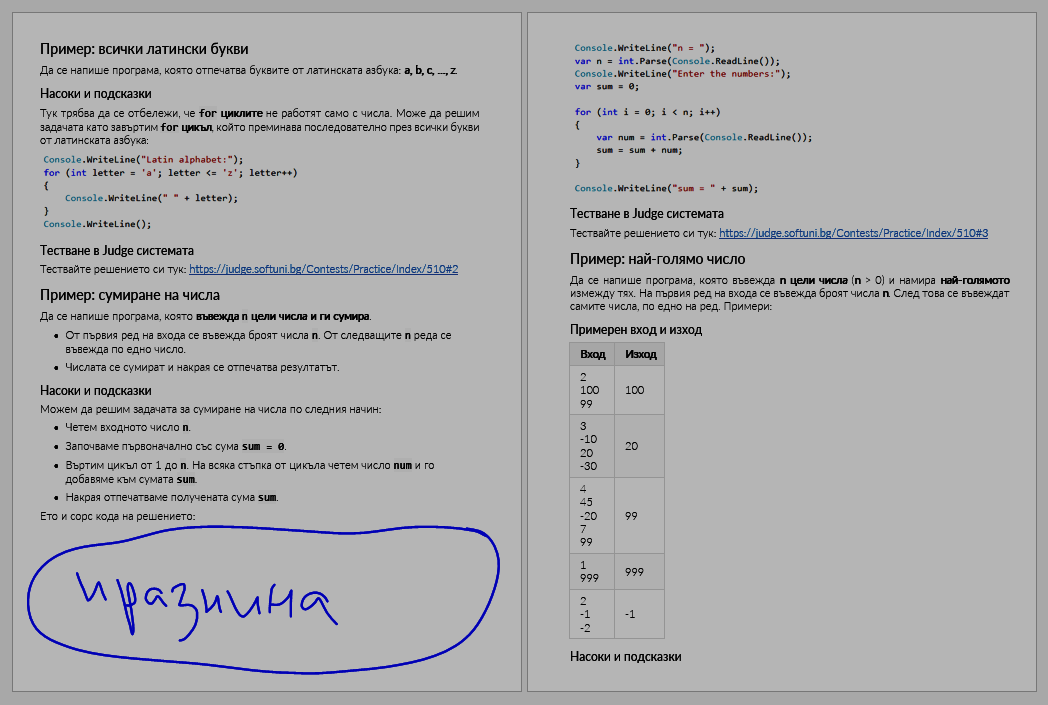


Може **да се редактира текста** или линка, за да се побере:

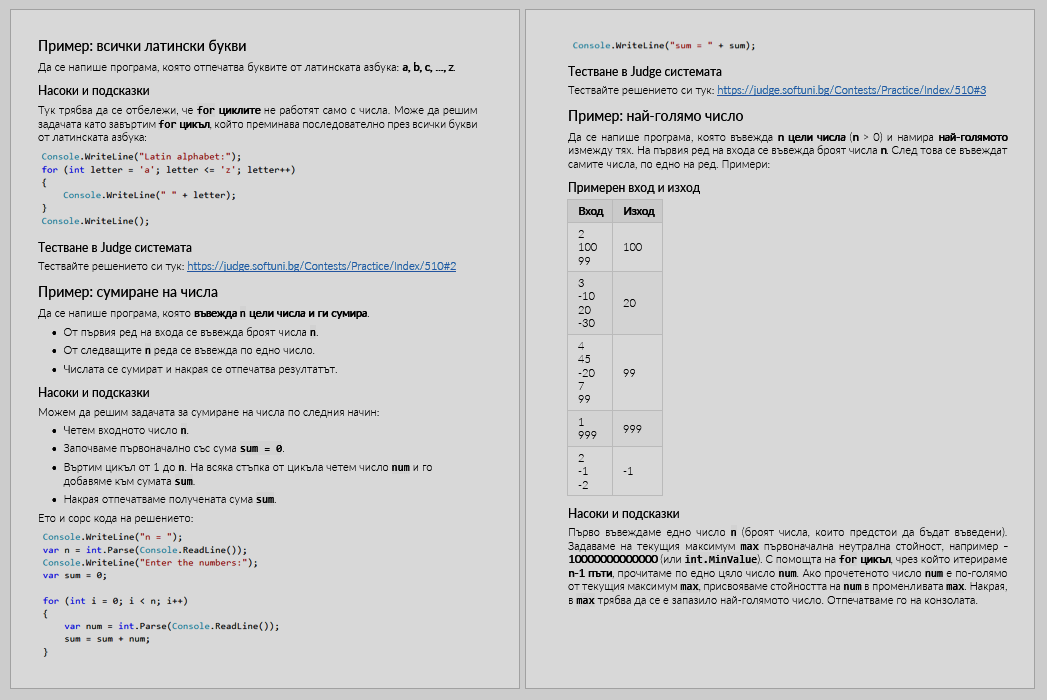


#### Премахване на празнини, причинени от картинки

Когато имаме картинка, която не се събира на текущата страница, тя причинява „**празнина**“:



Празнината може да се запълни като **се сцепи на две части картинката** със сорс кода:



#### Корица – предна и задна

* Добавят се картинките са предната и задната корица.
* Слагат се страниците преди същината на книгата (автори, реклами, други).
* Обратната страна на предната и задната корици се оставят празни.

#### Таблица със съдържание

* Вмъква се таблица със съдържание.

#### Реклами

* Когато дадена глава свършва на нечетна страница (дясна), следващата **четна (лява) страница** се запълва с **реклама**. Така всяка нова глава започва от нечетна (дясна) страница.