# Решения на задачите от глава 1 на книгата "Въведение в програмирането със C#"

Предлагаме ви решения на задачите от [книгата "Въведение в програ­мирането със C#"](http://www.introprogramming.info), заедно с анализ на задачата, описание на използва­ните идеи, алгоритми, подходи за решаване и тестове.

## Авторски колектив

**Георги Ваклинов**

Контакти: ([georgi.vaklinov@gmail.com](mailto:georgi.vaklinov@gmail.com))

**Георги Матейчев Матеев**

Контакти: ([gmmateev92@abv.bg](mailto:gmmateev92@abv.bg))

**Доротея Валентинова Агайна**

Контакти: ([doroteya\_agayna@yahoo.com](mailto:doroteya_agayna@yahoo.com))

**Стелян Добринов Стоянов**

Контакти: ([steliyan@gmail.com](mailto:steliyan@gmail.com))

|  |  |
| --- | --- |
| Задача 1. Запознаване със средата за разработка и помощната документация | |
| **Условие**  Запознайте се с Microsoft Visual Studio, Microsoft Developer Network (MSDN) Library Documentation. Инсталирайте си Visual Studio в къщи | |
| **Описание на входа** | |
| **Описание на изхода** | |
| **Анализ на задачата**  Към днешна дата актуалната версия на Microsoft Visual Studio е Visual Studio 2010. Информация за нея и характеристиките на инструментите, които включва можете да намерите на:  <http://www.microsoft.com/visualstudio/en-us>  Visual Studio Express е безплатният пакет, който предлага Microsoft и подробности за компонентите му можете да прочетете на:  <http://www.microsoft.com/visualstudio/en-us/products/2010-editions/express>  За целите на този курс е достатъчно да инсталирате и използвате Visual C# Express. Ще го намерите на:  <http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-csharp-express>  MSDN Library е основният и крайно необходим on-line източник на информация за продуктите, инструментите, технологиите и услугите предлагани от Microsoft. Библиотеката предлага подробна документация, примерен код, технически статии и т.н. Намира се на:  <http://msdn.microsoft.com/library/default.aspx> | |
| **Решение (сорс код)** | |
|  | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 2. Запознаване с конзолата | |
| **Условие**  Да се намери описанието на класа System.Console в стандартната .NET API документация (MSDN Library). | |
| **Описание на входа** | |
| **Описание на изхода** | |
| **Анализ на задачата**  Конзолата е част от стандартния вход и изход на конзолните приложения. Класът System.Console не може да се наследява. Информацията за свойствата и методите му можете да намерите на:  <http://msdn.microsoft.com/en-us/library/system.console.aspx> | |
| **Решение (сорс код)** | |
|  | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 3. Извеждане на текст в конзолата | |
| **Условие**  Да се намери описанието на метода System.Console.WriteLine() с различ­ните негови възможни параметри в MSDN Library. | |
| **Описание на входа** | |
| **Описание на изхода** | |
| **Анализ на задачата**  Методът System.Console.WriteLine() извежда текст на стандартния изход и като добавя и стандартния стринг за прекъсване на ред. Подробности може да намерите на:  <http://msdn.microsoft.com/en-us/library/zdf6yhx5.aspx>  Методът има 19 разновидности, които са описани на:  <http://msdn.microsoft.com/en-us/library/system.console.writeline.aspx>  На този етап е добре да се запознаете и с метода System.Console.ReadLine():  <http://msdn.microsoft.com/en-us/library/system.console.readline.aspx> | |
| **Решение (сорс код)** | |
|  | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 4. Изпробване на различните начини за компилация | |
| **Условие**  Да се компилира и изпълни примерната програма от примерите в тази глава през командния ред (конзолата) и с помощта на Visual Studio. | |
| **Описание на входа** | |
| **Описание на изхода**  Изходните данни трябва да се отпечатат на конзолата.  Единственият изходен ред трябва да се състои от поздравителното съобщение „Hello C#!”. | |
| **Анализ на задачата**  За да решите задачата е необходимо предварително да сте избрали среда за разработка, да сте я инсталирали и да я стартирате. Ще разгледаме случая, в който избраната среда е Visual Studio:   * Ако го стартирате за първи път ще трябва да направите някои допълнителни настройки, които в последствие могат да бъдат променяни; * Създайте нов проект от тип Console Application (задайте му подходящо име); * Добра практика е да дадете на Program.cs ново, по-подходящо име (HelloCSharp.cs например); * В Main метода на отворения за редактиране файл копирайте кодът от стр. 82; * Последната стъпка е да изберете Build Solution от менюто Build или да натиснете F6 от клавиатурата;   Ако в процеса на компилация са възникнали грешки, информация за тях ще намерите в прозореца Error List. Ако той е празен, значи кодът успешно е бил компилиран.  След отстраняването на евентуалните грешки и ако повторната компилация е била успешна можете да стартирате програмата с натискането на Start Without Debugging от менюто Debug или на Ctrl + F5. | |
| **Решение (сорс код)** | |
| class HelloCSharp  {  static void Main(string[] args)  {  System.Console.WriteLine("Hello C#!");  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 5. Поздрав | |
| **Условие**  Да се модифицира примерната програма, така че да изписва различно поздравление, например "Добър ден!". | |
| **Описание на входа** | |
| **Описание на изхода**  Изходните данни трябва да се отпечатат на конзолата.  Единственият изходен ред трябва да се състои от избраното от вас поздравително съобщение. | |
| **Анализ на задачата**   * Стартирайте във Visual Studio създаденият от предходната задача проект или изпълнете наново стъпките описани в нея; * Отворете за редакция файла HelloCSharp.cs; * В него намерете реда, който съдържа „Hello C#!”; * Променете текста в кавичките на избрания от вас поздрав; * Ако проектът се компилира успешно го стартирайте; | |
| **Решение (сорс код)** | |
| class HelloCSharp  {  static void Main(string[] args)  {  System.Console.WriteLine("Good morning, sunshine!");  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 6. Приятно ми е да се запознаем (My name is…) | |
| **Условие**  Напишете програма, която изписва вашето име и фамилия на конзолата. | |
| **Описание на входа** | |
| **Описание на изхода**  Изходните данни трябва да се отпечатат на конзолата.  Единственият изходен ред трябва да се състои от двете ви имена разделени с един интервал. | |
| **Анализ на задачата**   * Във Visual Studio създайте нов проект (не забравяйте да му зададете подходящо име); * Сменете името на Program.cs с подходящо за проекта; * Намерете Main метода; * В тялото му използвайте метода System.Console.WriteLine(); * Като негов параметър в двойни кавички подайте името и фамилията си; * Опитайте на компилирате така написания код; * Ако има грешки отстранете ги изпълнете предходната стъпка отново; * Стартирайте програмата с Ctrl + F5; | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace MyNameIs  {  class FirstLastName  {  static void Main(string[] args)  {  Console.WriteLine("Peter Petrov");  }  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 7. Двоичен поздрав | |
| **Условие**  Напишете програма, която извежда на конзолата числата 1, 101, 1001 на нов ред. | |
| **Описание на входа** | |
| **Описание на изхода**  Изходните данни трябва да се отпечатат на конзолата.  Единственият изходен ред трябва да се състои от дадените по-горе числа във формат **число, число, число**. | |
| **Анализ на задачата**   * Във Visual Studio създайте нов проект (не забравяйте да му зададете подходящо име); * Сменете името на Program.cs с подходящо за проекта; * Намерете Main метода; * В тялото му използвайте метода System.Console.WriteLine(); * Като негов параметър в двойни кавички изпишете зададените числа; * Опитайте на компилирате така написания код; * Ако има грешки отстранете ги изпълнете предходната стъпка отново; * Стартирайте програмата с Ctrl + F5; | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace NumberSequence  {  class NumberSequence  {  static void Main(string[] args)  {  Console.WriteLine("1, 101, 1001");  }  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 8. Часът е точно… | |
| **Условие**  Напишете програма, която извежда на конзолата текущата дата и час | |
| **Описание на входа** | |
| **Описание на изхода**  Изходните данни трябва да се отпечатат на конзолата.  Единственият изходен ред трябва да се състои от текущата дата и час. | |
| **Анализ на задачата**   * Във Visual Studio създайте нов проект (не забравяйте да му зададете подходящо име); * Сменете името на Program.cs с подходящо за проекта; * Намерете Main метода; * В тялото му използвайте метода System.Console.WriteLine(); * Като негов параметър подайте свойството Now на класа DateTime; * Опитайте на компилирате така написания код; * Ако има грешки отстранете ги изпълнете предходната стъпка отново; * Стартирайте програмата с Ctrl + F5; | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace CurrentDateTime  {  class CurrentDateTime  {  static void Main(string[] args)  {  Console.WriteLine(DateTime.Now);  }  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 9. Корен квадратен | |
| **Условие**  Напишете програма, която извежда корен квадратен от числото 12345. | |
| **Описание на входа** | |
| **Описание на изхода**  Изходните данни трябва да се отпечатат на конзолата.  Единственият изходен ред трябва да се състои от квадратния корен на даденото число. | |
| **Анализ на задачата**  Преди да започнете да решавате задачата намерете и прочетете информация за статичния клас System.Math и неговите свойства (<http://msdn.microsoft.com/en-us/library/system.math.aspx>).  Решението преминава през следните стъпки:   * Във Visual Studio създайте нов проект (не забравяйте да му зададете подходящо име); * Сменете името на Program.cs с подходящо за проекта; * Намерете Main метода; * В тялото му използвайте метода System.Console.WriteLine(); * Като негов параметър подайте изчисленият с помощта на класа Math корен квадратен от числото; * Опитайте на компилирате така написания код; * Ако има грешки отстранете ги изпълнете предходната стъпка отново;   Стартирайте програмата с Ctrl + F5; | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace SquareRoot  {  class SquareRoot  {  static void Main(string[] args)  {  Console.WriteLine(Math.Sqrt(12345));  }  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 10. Редица първа | |
| **Условие**  Напишете програма, която извежда първите 100 члена на редицата 2, -3, 4, -5, 6, -7, 8 | |
| **Описание на входа** | |
| **Описание на изхода**  Изходните данни трябва да се отпечатат на конзолата.  На един непрекъснат ред изпишете необходимият брой членове на редицата. | |
| **Анализ на задачата**  Преди да започнете да решавате задачата намерете и прочетете информация за метода System.Console.Write(), за условните оператори и циклите в C# или разгледайте следните линкове:   * За System.Console.Write() - <http://msdn.microsoft.com/en-us/library/system.console.write.aspx> * Условни оператори - <http://msdn.microsoft.com/en-us/library/5011f09h(v=VS.71).aspx> * Цикли - <http://msdn.microsoft.com/en-us/library/ch45axte.aspx>   Решението преминава през следните стъпки:   * Във Visual Studio създайте нов проект (не забравяйте да му зададете подходящо име); * Сменете името на Program.cs с подходящо за проекта; * Намерете Main метода; * В тялото му използвайте един **for** цикъл; * За всяка стойност на итериращата променлива проверявайте дали е четна или нечетна. Ако е четна отпечатайте на конзолата стойността й (с добавен интервал след нея). Ако е нечетна отпечатайте на конзолата отрицателната й стойност (отново с интервал след нея); * Опитайте на компилирате така написания код; * Ако има грешки отстранете ги изпълнете предходната стъпка отново; * Стартирайте програмата с Ctrl + F5; | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace NumberSequence  {  class NumberSequence  {  static void Main(string[] args)  {  for (int i = 2; i < 103; i++)  {  if (i % 2 == 0)  {  Console.Write(i + " ");  }  else  {  Console.Write(-i + " ");  }  }  Console.WriteLine();  }  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 11. Колко голям ще порасна? | |
| **Условие**  Направете програма, която прочита от конзолата вашата възраст и изписва (също на конзолата) каква ще бъде вашата възраст след 10 години. | |
| **Описание на входа**  Входът ще се изчита от конзолата.  На единственият ред ще се подава **N** (предполагаемата възраст на потребителя).  **N** винаги ще бъде цяло положително число между 0 и 150. | |
| **Описание на изхода**  Изходните данни трябва да се отпечатат на конзолата.  Единственият изходен ред трябва да съдържа бъдещата възраст на потребителя. | |
| **Анализ на задачата**  Преди да пристъпите към решаване на задачата се запознайте се с методa byte.Parse() (<http://msdn.microsoft.com/en-us/library/system.byte.parse.aspx>)  Решението преминава през следните стъпки:   * Във Visual Studio създайте нов проект (не забравяйте да му зададете подходящо име); * Сменете името на Program.cs с подходящо за проекта; * Намерете Main метода; * В тялото му декларирайте променлива от тип byte и й присвоете стойност с помощта на методите byte.Parse() и System.Console.ReadLine(); * Увеличете стойността на променливата с 10 и запазете резултата в същата променлива; * Отпечатайте новата стойност на конзолата (System.Console.WriteLine() с параметър името на променливата); * Опитайте на компилирате така написания код; * Ако има грешки отстранете ги изпълнете предходната стъпка отново;   Стартирайте програмата с Ctrl + F5; | |
| **Решение (сорс код)** | |
| using System;  using System.Collections.Generic;  using System.Linq;  using System.Text;  namespace NYearsOld  {  class NYearsOld  {  static void Main(string[] args)  {  byte age = byte.Parse(Console.ReadLine());  age += 10;  Console.WriteLine(age);  }  }  } | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 12. Разлики между C# и .NET Framework | |
| **Условие**  Опишете разликите между C# и .NET Framework. | |
| **Описание на входа**  Потърсете в Интернет какво представлява едното и какво другото. За начало можете да ползвате следните линкове:   * За C# - <http://msdn.microsoft.com/library/z1zx9t92> * За .NET Framework - <http://msdn.microsoft.com/library/zw4w595w.aspx> | |
| **Описание на изхода** | |
| **Анализ на задачата** | |
| **Решение (сорс код)** | |
|  | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 13. Популярни програмни езици | |
| **Условие**  Направете списък с най популярните програмни езици. С какво те се различават от C#? | |
| **Описание на входа** | |
| **Описание на изхода** | |
| **Анализ на задачата**  В сравнителна таблица подредете 10-те най-популярни езика и техните сравнителни характеристики, след което попълнете таблицата. | |
| **Решение (сорс код)** | |
|  | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |

|  |  |
| --- | --- |
| Задача 14. Декомпилация | |
| **Условие**  Да се декомпилира примерната програма от задача 5. | |
| **Описание на входа** | |
| **Описание на изхода** | |
| **Анализ на задачата**  Има няколко добри варианта за изпълнението на тази задача:   * JustDecompile на Телерик - <http://www.telerik.com/products/decompiler.aspx> * ILSpy - <http://wiki.sharpdevelop.net/ILSpy.ashx> * Reflector - <http://www.reflector.net/> * FileDisassembler - <http://www.denisbauer.com/NETTools/FileDisassembler.aspx> * DevExtras CodeReflect - <http://www.devextras.com/decompiler/> * JetBrains DotPeek - <http://www.jetbrains.com/decompiler/> | |
| **Решение (сорс код)** | |
|  | |
| **Тестове** | |
| **Вход** | **Изход** |
|  |  |
| **Вход** | **Изход** |
|  |  |