



Методический материал

# Пишем чат-бота

C# для новичков

# С# для новичков: пишем чат-бота

Говорите «программист» и представляете гения с высшим техническим образованием и врождёнными способностями? Времена изменились. Теперь создавать программы может каждый, а современные языки, такие как С#, призваны вам помочь.

## 3 мифа о программировании на С#

1

### Миф

Чтобы стать С#-разработчиком, надо учиться долгие годы.

### Правда

Достаточно и 6 месяцев. С# — один из основных языков в энтерпрайз-разработке, а это значит, что корпорации всегда нанимают стажёров и даже готовы сами их обучать. Что говорить про крепких джунов, которые уже много умеют.

2

### Миф

Чтобы быть С#-разработчиком, надо разбираться в высшей математике.

## Правда

Если вы хотите создавать программы для научных вычислений, то без математики вам никуда. Обычному разработчику, который пишет сайты, мобильные приложения или видеоигры, достаточно базовых знаний.

3

## Миф

C# — это сложный язык. Говорят, есть и варианты полегче.

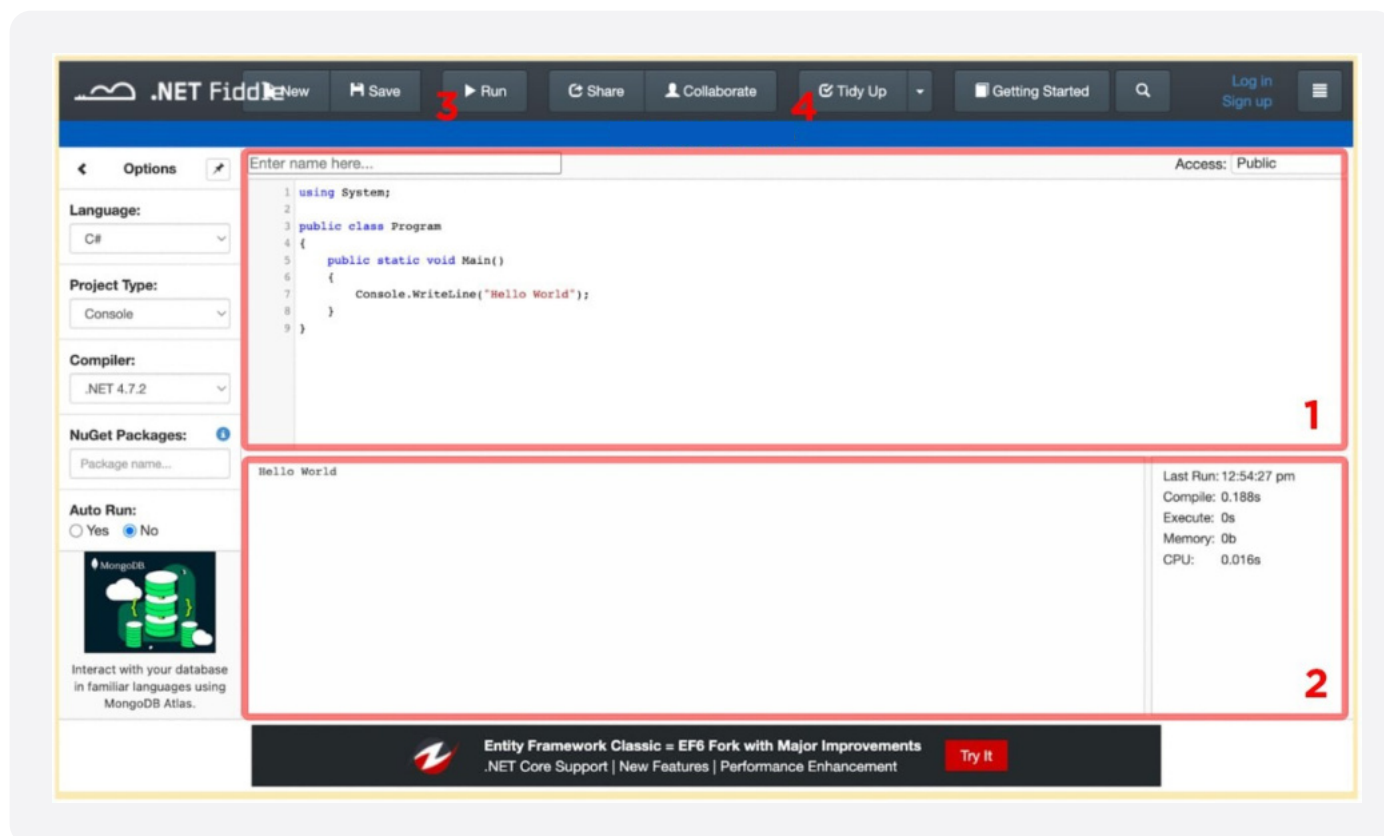
## Правда

C# изначально задумывался как язык, в котором нет ничего лишнего. Тут полно «синтаксического сахара» — простых конструкций, которые делают сложные вещи. Добавьте к этому хорошую читаемость кода — посмотрите на функцию, и сразу понятно, за что она отвечает.

Но это всё слова. Сейчас вы попробуете C# в деле и лично убедитесь, что в программировании нет ничего сложного.

## Пишем простой чат-бот

Зайдите на сайт [dotnetfiddle.net](https://dotnetfiddle.net). Это онлайн-среда для выполнения программ на C#. Да, большое приложение там не напишешь, но пока вы только пробуете язык, это лучший вариант.



- 1 Окно редактора. Здесь вы будете писать код
- 2 Консоль. Это экран, в котором будет выполняться ваша программа
- 3 Кнопка Run запускает код
- 4 Кнопка Tidy Up чистит форматирование, чтобы код было легче читать

Любая программа — это набор инструкций для компьютера или алгоритм. В случае чат-бота его можно описать так:

- 1 поприветствовать пользователя
- 2 Попросить пользователя задать вопрос и вывести его в консоль
- 3 Проверить, может ли программа ответить на вопрос
- 4 Если да, вывести ответ в консоль

Звучит несложно. Теперь переведём слова в код. Перепишите фрагмент кода ниже и вставьте его в окно онлайн-редактора:

```
using System;

public class Program
{
    public static void Main()
    {
        Console.WriteLine("Приветствую, задавай свой вопрос!");
        var text = Console.ReadLine();
        Console.WriteLine("Ваш вопрос: " + text);
        if (text == "Привет")
        {
            Console.WriteLine("Бот: Здорова!");
        }
        /*здесь ваш код*/
    }
}
```

Но это всё слова. Сейчас вы попробуете C# в деле и лично убедитесь, что в программировании нет ничего сложного.

## Разбираем код

### 1 Поприветствовать пользователя

```
Console.WriteLine("Приветствую, задавай свой вопрос!");
```

Здесь вы обращаетесь к консоли и «просите» её вывести текст в скобках. Фразу нужно взять в кавычки — без них программа не поняла бы, что перед ней человеческое приветствие, а не код.

Но это всё слова. Сейчас вы попробуете C# в деле и лично убедитесь, что в программировании нет ничего сложного.

Обратите внимание — каждая строка кода в C# заканчивается точкой с запятой.

## 2 Попросить пользователя задать вопрос и вывести его на экран

```
var text = Console.ReadLine();
```

Вновь обращаетесь к консоли, но уже с другой «просьбой». **ReadLine** значит «прочитай вопрос, который ввёл пользователь». Но у компьютера память короткая. Поэтому только прочитать вопрос мало, его нужно сохранить.

Куда сохранять? В переменную **text**, которая объявлена с помощью ключевого слова **var**.

### Что за переменная?

Это область в памяти компьютера. Представьте шкафчик с ящиками. В каждый можно положить только одну вещь и присвоить ему номер, чтобы потом было легче найти. То же самое и в компьютере. Ящики — это переменные, в которые помещается информация и маркируется понятным именем.

Теперь вопрос выводится на экран:

```
Console.WriteLine("Ваш вопрос: " + text);
```

Уже знакомая вам конструкция, но кое-что интересное внутри скобок. Текст не просто выводится на экран, а «соединяется» с переменной, в которой хранится вопрос пользователя. По-программистки это называется словом «конкатенация».

Обратите внимание — каждая строка кода в C# заканчивается точкой с запятой.

### 3 Проверить, может ли программа ответить на вопрос.

```
if (text == "Привет")
```

Помните, что в C# достаточно только взглянуть на код, и сразу понятно, что он делает? Это тот самый случай. If по-английски «если».

Прописывается условие: если переменная `text` содержит фразу «Привет», то необходимо сделать что-то. Напомним, что в переменной (ящике) `text` хранится вопрос пользователя.

### 4 Вывести ответ в консоль

Код, который нужно выполнить, если условие из предыдущего шага верно:

```
{  
    Console.WriteLine("Бот: Здравова!");  
}
```

Обратите внимание — этот фрагмент заключается в фигурные скобки. Сам код вам уже до боли знаком — вновь «просите» консоль вывести строку.

Теперь вы можете расширить возможности своего бота и «научить» его отвечать на другие вопросы. Удалите комментарий **/\*здесь ваш код\*/** и добавьте на его место ещё несколько условий.

Например, вопрос про погоду:

```
if (text == "Как погода?")
{
    Console.WriteLine("Бот: У природы нет плохой погоды!");
}
```



У вас получилось вот так:

```
using System;
```

```
public class Program
```

```
{
```

```
    public static void Main()
```

```
    {
```

```
        Console.WriteLine("Приветствую, задавай свой вопрос!");
```

```
        var text = Console.ReadLine();
```

```
        Console.WriteLine("Ваш вопрос: " + text);
```

```
        if (text == "Привет")
```

```
        {
```

```
            Console.WriteLine("Бот: Здравова!");
```

```
        }
```

```
        if (text == "Как погода?")
```

```
        {
```

```
            Console.WriteLine("Бот: У природы нет плохой погоды!");
```

```
        }
```

```
        if (text == "Я выучу C#?")
```

```
        {
```

```
            Console.WriteLine("Бот: Обязательно! Главное - не забывай  
            практиковаться");
```

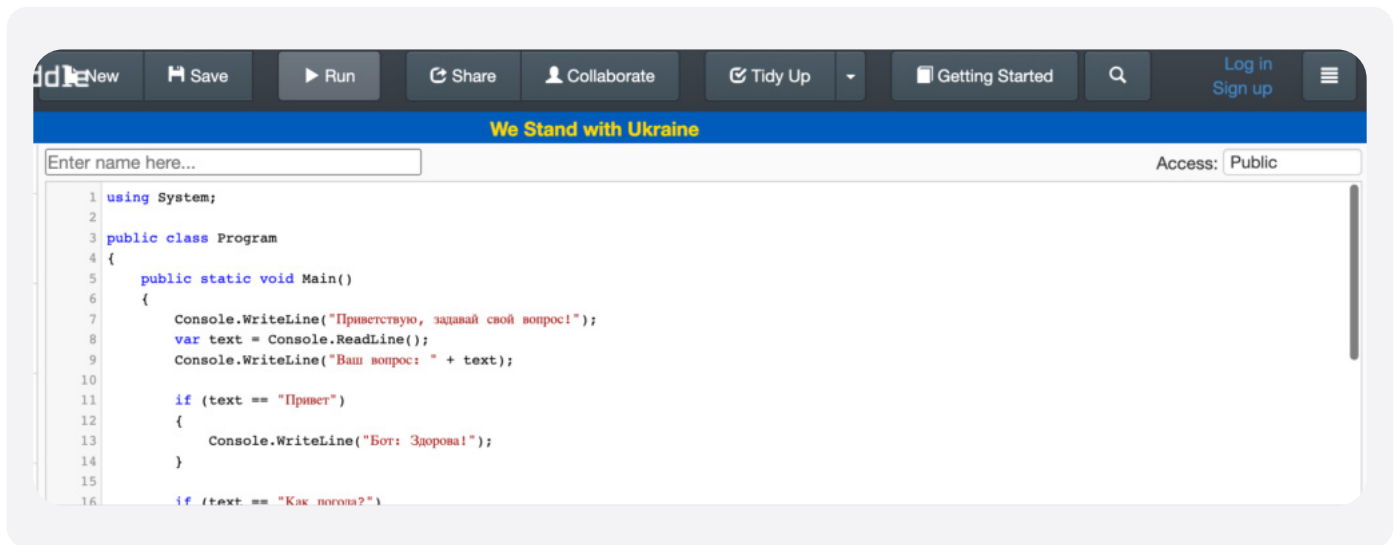
```
        }
```

```
    }
```

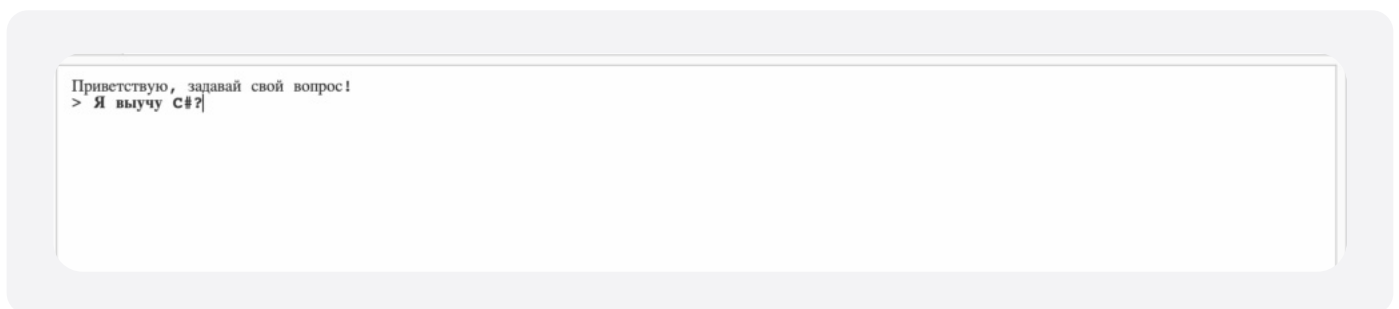
```
}
```

# Запуск программы

- 1 Кликните **Run**, чтобы запустить код

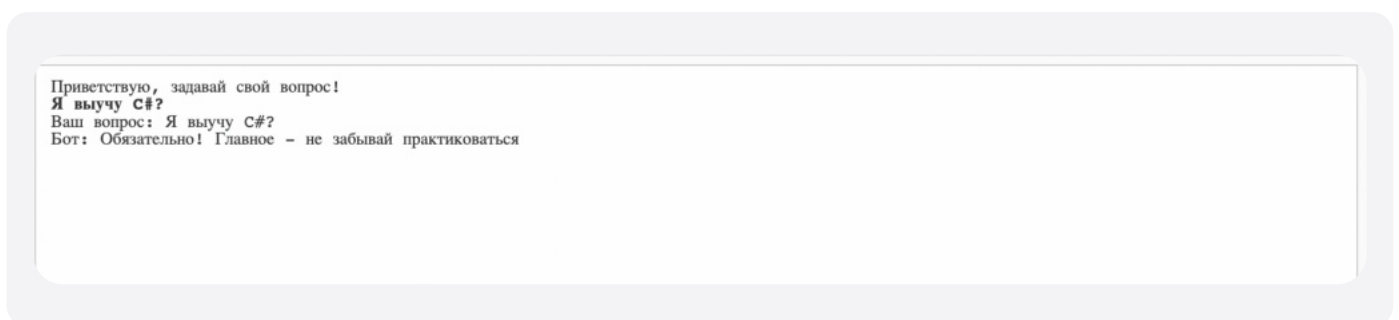


- 2 Введите в консоль один из вопросов, на которые вы «учили» отвечать своего бота



- 3 Нажмите **Enter**

- 4 Если в коде нет ошибок, бот напечатает вам ответ



Теперь вы знаете, как отобразить текст, запросить данные у пользователя и работать с условиями. Как видите, чтобы писать программы, не нужно выполнять сложные расчёты и владеть какими-то особыми знаниями — нужно лишь немного практики. А когда вам помогает опытный разработчик, научиться кодить — лишь вопрос времени.

На курсе «Профессия C#-разработчик» вас будет сопровождать проверяющий эксперт — он ответит на вопросы, проведёт код-ревью и даст полезные советы из своего опыта. Вы добавите в портфолио проекты коммерческого уровня: разработаете мессенджер в команде, CRM-систему, лендинг и интернет-магазин. А Центр карьеры Skillbox поможет как можно быстрее найти работу ⚡