Главное управление образования Гомельского облисполкома

Учреждение образования

«Гомельский государственный машиностроительный колледж»

Учебная дисциплина

«Конструирование программ и языки программирования»

**Инструкция**

по выполнению лабораторной работы №4

«Разработка программ с использованием методов»

Гомель 2018

Составитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Гаврилова В.М.

Обсуждено и одобрено на заседании цикловой комиссии «Программное обеспечение информационных технологий»

Протокол от «01» сентября 2018 № 1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.Ю. Лавская

**Лабораторная работа №4**

**Тема работы: «Разработка программ с использованием методов»**

1. **Цель работы**

Научиться разработке и перегрузке методов с параметрами различного статуса, использованию встроенных методов.

1. **Задание**

**Задание 1**. Написать программу, реализующую перегрузку методов согласно варианту задания для значений переменных целого, вещественного и строкового типов. Исходные данные вводятся с клавиатуры.

1. Реализовать метод возведения в квадрат суммы двух значений.
2. Реализовать метод возведения в квадрат разности двух значений.
3. Реализовать метод возведения в квадрат произведения трех значений.
4. Реализовать метод возведения в квадрат частного двух значений.
5. Реализовать метод возведения в куб значений различных типов.
6. Реализовать метод возведения в куб суммы трех значений.
7. Реализовать метод возведения в куб разности трех значений.
8. Реализовать метод возведения в куб произведения двух значений.
9. Реализовать метод возведения в куб частного двух значений.
10. Реализовать метод нахождения суммы кубов двух значений.
11. Реализовать метод нахождения разности кубов двух значений.
12. Реализовать метод нахождения минимального из трех значений.
13. Реализовать метод нахождения частного двух значений.
14. Реализовать метод вычисления суммы квадратов трех значений.
15. Реализовать метод вычисления разности двух значений.
16. Реализовать метод вычисления частного двух значений.
17. Выполните перегрузку методов для операции вычитания.
18. Выполните перегрузку методов для операции деления.
19. Выполните перегрузку методов для операции умножения.
20. Реализовать метод для возведения значения переменной в квадрат.
21. Реализовать метод возведения в квадрат суммы двух значений.
22. Реализовать метод возведения в квадрат разности двух значений.
23. Выполните перегрузку методов для операции сложения.

**Задание 2.** Написать программу, реализующую функцию согласно варианту задания.

1. Даны два натуральных числа. Выяснить, в каком из них больше цифр. Определить метод для расчета количества цифр натурального числа.
2. Получить все шестизначные счастливые номера. Счастливым называют такое шестизначное число, в котором сумма его первых трех цифр равна сумме его последних трех цифр. Определить метод для расчета суммы цифр трехзначного числа.
3. Даны два натуральных числа. Выяснить, является ли хоть одно из них палиндромом, т.е. таким числом, десятичная запись которого читается одинаково слева направо и справа налево. Определить метод, позволяющий распознавать числа-палиндромы.
4. Даны шесть различных чисел. Определить максимальное из них. Определить метод, находящий максимум из двух различных чисел.
5. Даны действительные числа *s* и *t*. Получить

где

1. Даны действительные числа *s* и *t*. Получить

где

1. Даны шесть различных чисел. Определить минимальное из них. Определить метод, находящий минимум из двух различных чисел.
2. Даны два натуральных числа. Выяснить, в каком из них произведение цифр больше. Определить метод для расчета произведения цифр натурального числа.
3. Найти все двузначные простые числа. Определить метод, позволяющий распознать простые числа.
4. Даны действительные числа *s* и *t*. Получить

где

1. Определить значение z = max(a,2b) \* max(2a-b,b), где max(x,y) – максимальное из чисел x,y. Написать метод для определения максимального значения.
2. Определить значение z = min(a,3b) \* min(2a-b,2b), где min(x,y) – минимальное из чисел x,y. Написать метод для определения минимального значения.
3. Определить значение z=sign*x* + sign*y*, где
4. Даны основания и высоты двух равнобедренных трапеций. Найти сумму их периметров. Написать метод для расчета периметра равнобедренной трапеции по ее основаниям и высоте.
5. Даны три квадратных уравнения сколько из них имеют вещественные корни. Определить метод, позволяющий распознавать наличие вещественных корней в квадратном уравнении.
6. Найти периметр треугольника, заданного координатами своих вершин. Определить метод для расчета длины отрезка по координатам его вершин.
7. Даны две последовательности целых чисел: и . Найти количество четных чисел в первой из них и количество нечетных во второй. Определить метод, позволяющий распознавать четные числа.
8. Найти все трехзначные простые числа. Определить метод, позволяющий распознать простые числа.
9. Найти значение выражения

где n! означает факториал числа n (n! = 1\*2\*..\*n). Определить метод для расчета факториала натурального числа.

1. Даны два натуральных числа. Выяснить, в каком из них сумма цифр больше. Определить метод для расчета суммы цифр натурального числа.
2. **Оснащение работы**

Технические средства обучения:

* IBM – совместимый компьютер;

Электронные средства обучения:

* MS Visual Studio;
* MS Office.

1. **Основные теоретические сведения**

Функции в языке С# чаще всего рассматриваются в качестве **методов** класса. Метод - это элемент (член) класса, который реализует любые действия, выполняемые классом. Методы определяют *поведение* класса.

Метод (функция) – это законченный фрагмент кода, к которому можно обратиться по имени. Он описывается один раз, а вызываться столько, сколько это необходимо. Один и тот же метод может обрабатывать различные данные, переданные ему в качестве аргументов.

Синтаксис **заголовка** метода:

[спецификаторы] тип возвращаемого значения имя\_метода ([параметры])

В качестве **спецификаторов** могут выступать спецификаторы void, virtual, extern, public,static и множество других.

**Тело метода** задает последовательность действий (последовательность операторов), выполняемых методом. **Тип возвращаемого значения** определяет – значение какого типа будет вычислено с помощью данного метода.

После выполнения метода происходит возврат в точку вызова метода, передача туда значения, записанного в операторе return. Если метод не возвращает никакого значения, в его заголовке задается тип void, а оператор return отсутствует.

**Параметры метода** представляют собой локальные переменные, которые при вызове метода принимают значение соответствующего аргумента. Область действия параметра – весь метод.

*Пример простейшего метода*

public double Func1(int x,double y) //заголовок метода

{ //тело метода

double z=x\*y;

return z;

} //конец тела метода

/\*public –спецификатор

double – тип возвращаемого значения

Func1 - имя метода

int x,double y – параметры метода.

В точку вызова данного метода будет передано вещественное произведение двух чисел.\*/

Метод, не возвращающий значение – вызывается как отдельный оператор.

Метод, возвращающий значение - вызывается в правой части оператора присваивания.

Параметры, описываемые в заголовке метода, определяют множество значений аргументов, которые можно передавать в метод. Список аргументов при вызове метода должен соответствовать списку параметров по количеству, порядку следования и типу. Для каждого параметра должны задаваться его тип и имя.

**Способы обмена информацией**

При вызове метода выполняются следующие действия:

1. Вычисляются выражения, стоящие на месте аргументов
2. Выделяется память под параметры метода в соответствии с их типом
3. Каждому из параметров сопоставляется соответствующий аргумент
4. Выполняется тело метода
5. Если метод возвращает значение, оно передается в точку вызова. Если метод имеет тип void управление передается на оператор, следующий после вызова.

*Пример передачи параметров*

namespace ConsoleApplicationl

{ class Classl

{

int Max(int a, int b) // a,b-**параметры метода**, int-возвращаемое значение

{

if (a > b) return a;

else return b; //возврат в точку вызова максимального из 2-х чисел

}

static void Main()

{

int a = 2, b = 4;

int x = Max(a, b); //первый вызов метода с **аргументами** a,b

Console.WriteLine(x);

short t1 = 3, t2 = 4;

int y = Max(t1, t2); //второй вызов метода

Console.WriteLine(y);

int z = Max(a + 3, 3 / 2 \* 4); //третий вызов метода

Console.WriteLine(z);} } }

В приведенном примере описан метод Max, который выбирает наибольшее из двух переданных ему значений. Параметрами метода являются a и b. В основной программе метод Max вызывается три раза. В первый раз методу передаются два аргумента того же типа, что и параметры, во второй раз – аргументы совместимого типа, в третий раз – выражение.

Количество аргументов должно соотстветствовать количеству параметров. На имена никаких ограничений не накладывается - могут совпадать или не совпадать.

Существует два способа передачи параметров:

1. по значению
2. по ссылке

При передаче *по значению* метод получает копии аргументов, и операторы внутри метода работают с этими копиями. Доступа к исходным значениям аргументов у метода нет, и, следовательно, нет возможности их изменять.

При передаче *по ссылке (по адресу)* метод получает адреса аргументов, осуществляет доступ к ячейкам памяти по этим адресам, следовательно может изменять значения аргументов, модифицируя параметры.

В языке C# для обмена данными между вызывающей и вызываемой функциями есть 4 типа параметров:

1. параметры-значения
2. параметры-ссылки
3. выходные параметры
4. параметры-массивы

**Параметры-значения**

Параметры-значения задаются в заголовке метода обычным образом, например:

void func(int x)

В данном примере параметр х – это параметр-значение, который предсталяет собой локальную переменную, которая получает свое значение при вызове метода func. В метод передается **копия** значения аргумента для х.

**Параметры-ссылки**

Во многих методах все величины, которые метод должен получить в качетстве исходных данных,описываются в списке перемтров, а величина, которую вычислет метод как результат своей работы, возвращается в вызывающий код с помощью оператора return. Если метод должен возвращать более,чем одно значение, такой способ не годится.

Также, если в методе требуется изменить значение передаваемых в него величин используются параметры –ссылки.

Параметры-ссылки задаются в заголовке метода с использованием ключевого слова ***ref***, например:

void func(**ref** int x)

При вызове такого метода параметр копирует не значение аргумента для х, а его адрес, поэтому внутри метода значение переданного параметра изменяется.

*Пример применения параметров-значений и параметров-ссылок*

namespace ConsoleApplication2

{

class Program

{

**static** void p(int a, ref int b)//обратите внимание на слово static!

{

a = 44;

b = 33;

Console.WriteLine("внутри метода {0} {1}", a, b);

}

static void Main(string[] args)

{

int a = 2, b = 4;

Console.WriteLine("до вызова метода {0} {1}", a, b);

**p(a, ref b);//обратите внимание на ВЫЗОВ метода**

Console.WriteLine("после вызова метода {0} {1}", a, b);

Console.ReadKey();

} }}

Если в методы передаются объекты (экземпляры классов), то есть величины не значимых, а ссылочных типов, то эти объекты всегда передаются по ссылке!

**Выходные параметры**

Часто возникает необходимость в методах, которые формируют несколько величин. Параметру, имеющему спецификатор out, должно быть присвоено значение внутри метода. Тем самым, можно из метода получить несколько значений-результатов работы.

**Модифицированный пример применения выходных параметров**

namespace ConsoleApplication2

{

class Program

{

static void p(int a, out int b, out int c)

{

a = 44;

b = 33;

c = a + b;

Console.WriteLine("внутри метода {0} {1} {2}", a, b,c);

}

static void Main(string[] args)

{

int a = 2, b,c;

p(a, out b,out c); **//обратите внимание на ВЫЗОВ метода**

Console.WriteLine("после вызова метода {0} {1} {2}", a, b,c);

Console.ReadKey();

} }}

Хорошим стилем программирования считается, если в списке параметров сначала записываются входные параметры, затем – параметры ссылки, затем – выходные параметры.

## Передача параметров типа значения

Переменная типа значения содержит данные непосредственно, в противоположность переменной ссылочного типа, которая содержит ссылку на данные. Поэтому передача переменной типа значения методу означает передачу методу копии переменной. Любые изменения параметра, выполняемые внутри метода, не влияют на исходные данные, хранимые в переменной. Если требуется, чтобы вызываемый метод изменял значение параметра, его следует передавать ссылкой с помощью ключевого слова **ref** или **out**.

### Передача типов значений с помощью значения

**Пример:** написать метод, вычисляющий факториал числа, и возвращающий это значение.

static int factorial(int n)

{

int i, res; res = 1;

for (i = 1; i <= n; i++)

res = res \* i;

return res;

}

static void Main(string[] args)

{

int i;//независимая переменная

for (i = 0; i <= 13;i++)

Console.WriteLine("{0}! = {1}",i,factorial(i));

Console.ReadKey();

}

### Передача типов значений с помощью ссылки

Ключевое слово **out** используется для передачи аргументов по ссылке. Оно похоже на ключевое слово **ref**, за исключением того, что **ref** требует инициализации переменной перед ее передачей. Для работы с параметром **out** определение метода и вызывающий метод должны явно использовать ключевое слово **out**. Таким образом, если значение параметра до вызова метода не определено, используют **out**, если определено используют **ref**.

**Параметры-массивы**

Иногда удобно создать метод, в который можно передавать разное количество аргументов. Язык C# предоставляет такую возможность с помощью ключевого слова *params.*

Параметр, помеченный этим ключевым словом, обозначает массив заданного типа неопределенной длины, например:

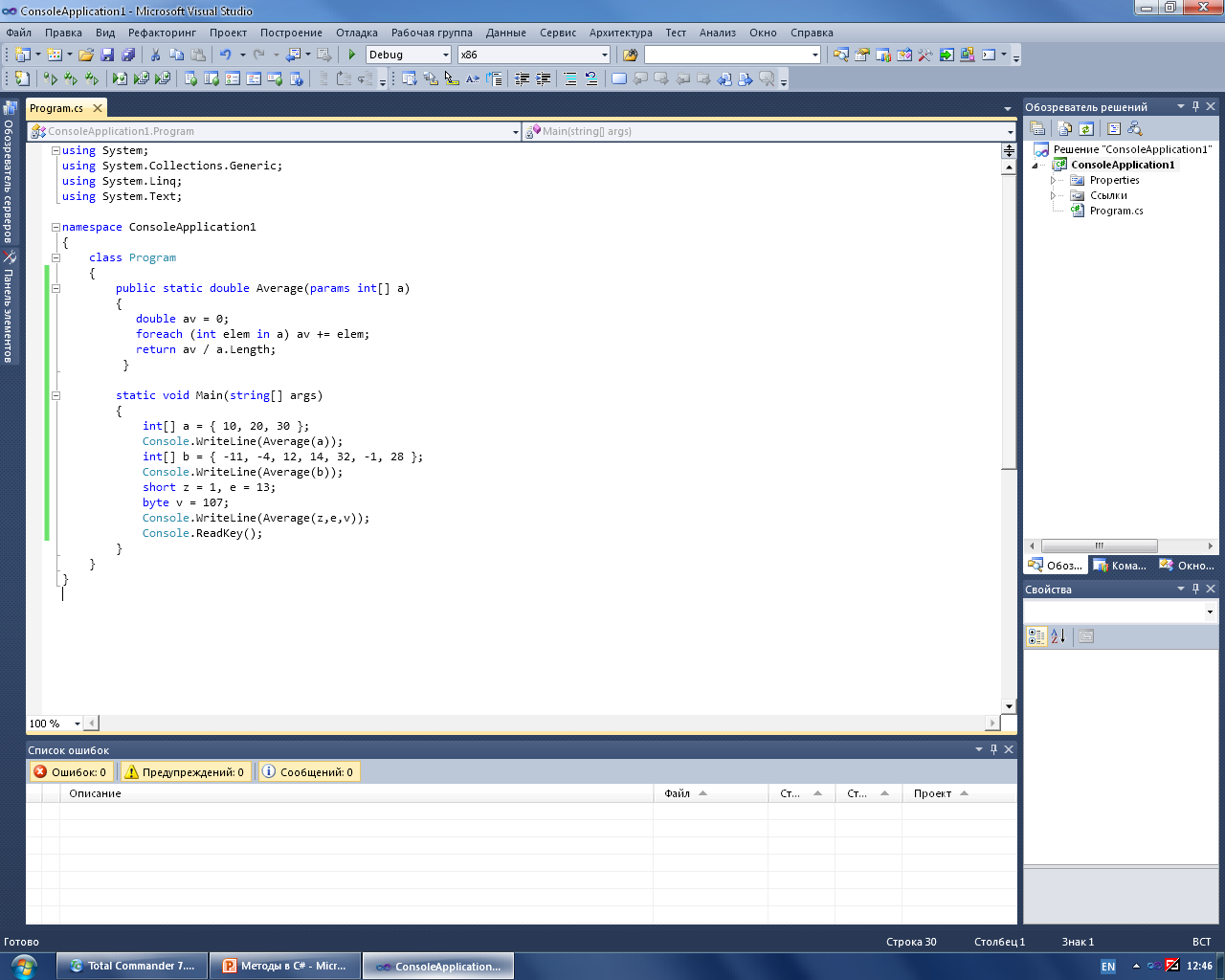
*public int Calculate(int a, out int c, params int[] d) …*

В этот метод можно передать три и более параметров. Внутри метода к параметрам, начиная с третьего, обращаются как к обычным элементам массива.

Количество элементов массива получают с помощью его свойства Length.

***Параметр-массив может быть только один и должен располагаться последним в списке.***

**Пример:** написать метод, вычисляющий среднее арифметическое различно количества элементов, принимаемых в качестве параметра .

****

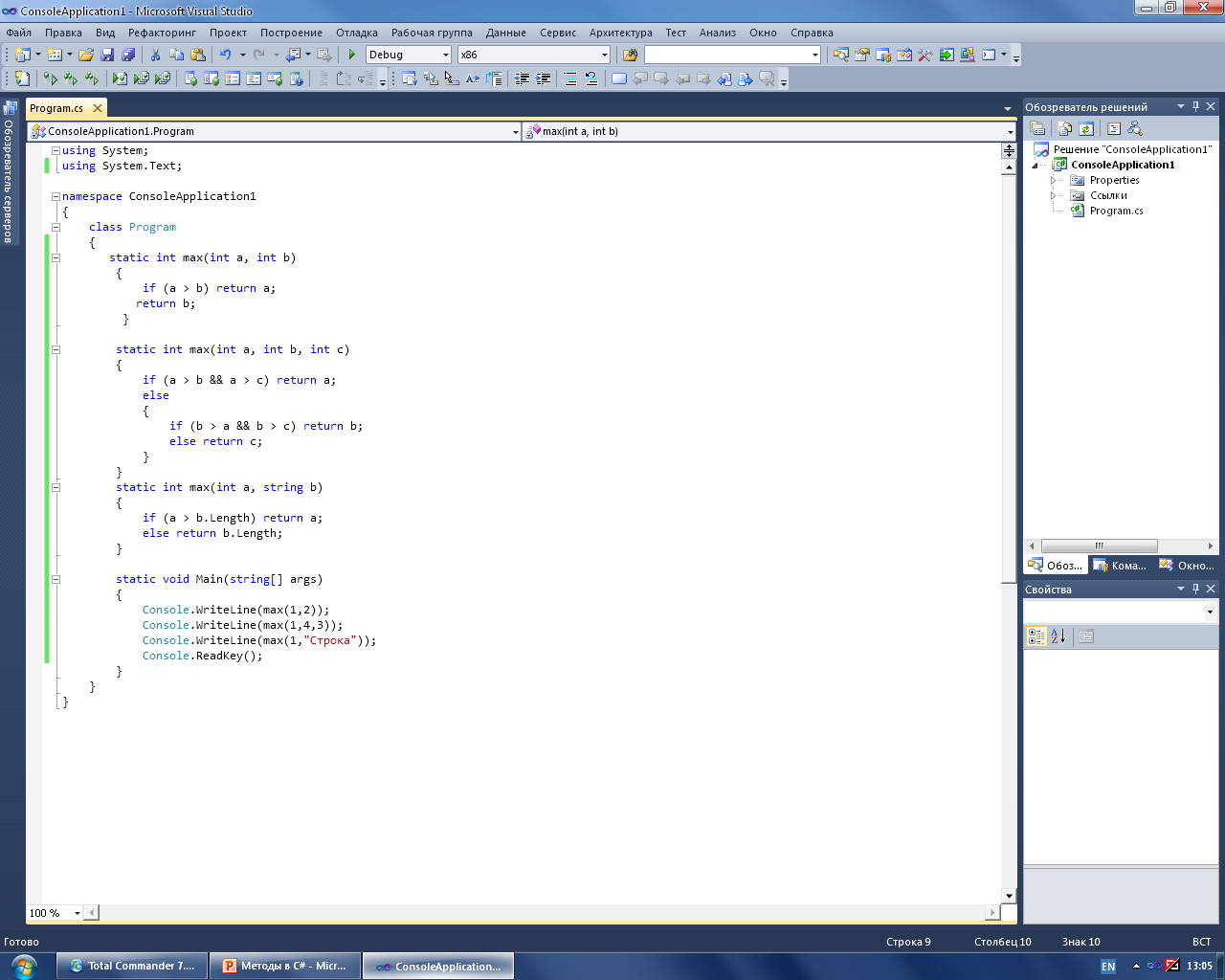
***Перегрузкой методов*** называется использование нескольких методов с одним и тем же именем, но различными типами параметров.

Компилятор определяет, какой именно метод требуется вызвать, по типу фактических параметров. Этот процесс называется ***разрешением (resolution)*** перегрузки.

Тип возвращаемого методом значения в разрешении не участвует.

Механизм разрешения основан на достаточно сложном наборе правил, смысл которых сводится к тому, чтобы использовать метод с наиболее подходящими аргументами и выдать сообщение, если такой не найдется.

**Пример.** Перегруженный метод для нахождения максимального из двух значений целого типа, трех целочисленных значений, а также значений целого и строкового типов.

****

1. **Порядок выполнения работы**

1 Разработайте алгоритм решения задач в виде блок-схемы

2 Создайте консольное приложение в среде Microsoft Visual Studio

3 Осуществите ввод данных в разрабатываемых программах

4 Выполните задание согласно варианту

5 Осуществите вывод результата на консоль

6 Оформите отчет о работе

1. **Форма отчета о работе**

*Номер учебной группы \_\_\_\_\_\_\_\_*

*Фамилия, инициалы обучающегося \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Дата выполнения работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Тема работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Цель работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Задание: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Оснащение работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*Результат выполнения работы: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1. **Контрольные вопросы и задания**
2. Дайте определение метода
3. Перечислите типы параметров, передаваемых в метод?
4. Что называют перегрузкой методов?
5. Каким образом можно перегрузить метод?

**Рекомендуемая литература**

**Албахари, Д.** C# 5.0. Справочник. Полное описание языка / Д. Албахари, Б. Албахари : [пер. с англ. Ю.Н.Артеменко]. М., 2013.

**Гриффитс, И.** Программирование на С# 5.0 / И. Гриффитс ; [пер. с англ. И.М. Райтмана]. М., 2014.

**Павловская, Т.А.** C#. Программирование на языке высокого уровня / Т.А. Павловская. СПб., 2015.

**Стиллмен, Э.** Изучаем С# / Э. Стиллмен, Дж. Грин. СПб., 2014.