



1920

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
филиал Федерального государственного бюджетного  
образовательного учреждения высшего образования  
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  
в г. Новороссийске  
Кафедра информатики и математики

УТВЕРЖДАЮ



Проректор по работе с филиалами  
ФГБОУ «Кубанский  
государственный университет»  
А.А.Евдокимов

2018 г.

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**  
**Б1.В.ДВ.08.02 ПРОГРАММИРОВАНИЕ НА С#**

Направление подготовки: 01.03.02 Прикладная математика и информатика  
Направленность (профиль): Системный анализ, исследование операций и управление (Математическое и информационное обеспечение экономической деятельности)

Программа подготовки: академическая

Форма обучения: очная

Квалификация (степень) выпускника: Бакалавр

Краснодар 2018

Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 01.03.02 Прикладная математика и информатика, утвержденного приказом Министерства образования и науки Российской Федерации № 228 от 12 марта 2015 года.

Программу составил(и):

И.Г.Рзун , доцент канд.физ.-мат.наук

С.В. Дьяченко доцент канд.физ.-мат.наук



Рабочая программа дисциплины Программирование на С# утверждена на заседании кафедры Информатики и математики протокол № 1 от 30.08. 2018 г.

Заведующий кафедрой (разработчик) Рзун И.Г.



Рабочая программа обсуждена на заседании кафедры Информатики и математики протокол № 1 от 30.08. 2018 г.

Заведующий кафедрой (выпускающей) Рзун И.Г.



Рабочая программа одобрена на заседании учебно-методической комиссии филиала УГС 01.00.00 «Математика и механика» 30.08. 2018 г. протокол № 1

Председатель УМК



С.В. Дьяченко

Рецензенты:

Сулимов А.В. Директор ООО «Центр компьютерной техники»

Посаженников А.В. Директор ООО «Профессиональные информационные технологии»

## **Содержание рабочей программы дисциплины**

- 1 Цели и задачи изучения дисциплины
  - 1.1 Цель освоения дисциплины
  - 1.2 Задачи дисциплиныОсновными задачами курса является:
  - 1.3 Место дисциплины в структуре образовательной программы
  - 1.4 Перечень планируемых результатов обучения по дисциплине
2. Структура и содержание дисциплины
  - 2.1 Распределение трудоёмкости дисциплины по видам работ
  - 2.2 Структура дисциплины:
  - 2.3 Содержание разделов дисциплины:
    - 2.3.1 Занятия лекционного типа
    - 2.3.2 Занятия практического типа
    - 2.3.3 Лабораторные занятия
    - 2.3.4 Примерная тематика курсовых работ (проектов)
  - 2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплинеПримерные вопросы для самостоятельной работы.
3. Образовательные технологии
4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации
  - 4.1 Фонд оценочных средств для проведения текущего контроля
  - 4.2 Фонд оценочных средств для проведения промежуточной аттестации
5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины
  - 5.1 Основная литература:
  - 5.2 Дополнительная литература:
  - 5.3. Периодические издания:
6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины
7. Методические указания для обучающихся по освоению дисциплины
8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)
  - 8.1 Перечень информационных технологий.
  - 8.2 Перечень необходимого программного обеспечения.
  - 8.3 Перечень информационных справочных систем:
9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.

## **1 Цели и задачи изучения дисциплины**

### **1.1 Цель дисциплины**

Целью дисциплины «Программирование на С#» является:

- изучение основных возможностей языка программирования С#, технологий структурного, модульного и объектно-ориентированного проектирования программ;
- овладение студентами практическими навыками написания и отладки программ;
- изучение современных методов визуального, объектно-ориентированного проектирования приложений, использующих в своей работе язык программирования С#.

При этом основное внимание необходимо уделить не рассмотрению максимально широкого круга вопросов, а на получение студентами глубоких знаний по фундаментальным основам информатики, на формирование у них общего информационного мировоззрения и на развитие алгоритмического мышления. Цели дисциплины соответствуют формируемой компетенции ОПК-3, ПК-7.

### **1.2 Задачи дисциплины**

- ознакомление с теоретическими основами программирования.
- изучение основ алгоритмизации.
- изучение средств описания данных.
- изучение средств описания действий языков программирования.
- овладение навыками программирования.
- освоение современных сред создания программных продуктов;
- способность к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям;
- способность критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности;
- способность к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения.

### **1.3 Место дисциплины в структуре образовательной программы**

Дисциплина «Язык программирования С#» относится к выборочной части учебного плана. Она является логически и содержательно - методически связана с такими дисциплинами как «Основы информатики», «Языки и методы программирования», «Компьютерный практикум».

Данная дисциплина базируется на следующих дисциплинах: «Основы информатики», «Системное программное обеспечение», «Языки и методы

программирования ”, “Базы данных”. Знания, полученные при освоении данной дисциплины, необходимы при выполнении выпускной квалификационной работы.

#### 1.4 Перечень планируемых результатов обучения по дисциплине , соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся компетенций ОПК-3, ПК-3, ПК-7

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
1.	ОПК-3	способностью к разработке алгоритмических и программных решений в области системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям	современные алгоритмы и программные продукты в области системного и прикладного программирования; нормативно-правовую базу по вопросам использования и создания программных продуктов и информационных ресурсов; понятие и назначение моделирования , этапы разработки математических, информационных и имитационных моделей; математические, информационные и имитационные модели, используемые в различных областях знаний;	разрабатывать системное и прикладное программное обеспечение для решения задач профессиональной деятельности разрабатывать математические, информационные и имитационные модели для решения задач профессиональной деятельности; разрабатывать информационные ресурсы глобальных сетей; решать педагогические задачи, связанные с поиском, хранением, обработкой и представлением информации; оценивать преимущества, ограничения и выбирать программные и аппаратные средства для решения профессиональн	навыками разработки алгоритмов и программ в области системного и прикладного программирования; навыками разработки математических, информационных и имитационных моделей для решения практических задач; навыками разработки информационных ресурсов глобальных сетей для решения практических задач; способами ориентирования и взаимодействия с ресурсами информационной образовательной среды,

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
			<p>современные интернет - технологии; процессы информатизации общества и образования; сущность и структуру информационных процессов в современной образовательной среде, типологии электронных образовательных ресурсов; базовые понятия в области построения баз данных и работы с ними; современные базы данных и системы управления базами данных. методологию испытаний и построения системы оценки качества систем и программных средств.</p>	<p>ых и образовательных задач; оценивать основные педагогические свойства электронных образовательных продуктов и определять педагогическую целесообразность их использования в учебном процессе проектировать и разрабатывать базы данных; разработать план тестирования систем и программных средств.</p>	<p>осуществления выбора различных моделей использования информационных и коммуникационных технологий в учебном процессе с учетом реального оснащения образовательного учреждения, совершенствования профессиональных знаний и умений путем использования возможностей информационной среды; навыками проектирования и разработки прикладных баз данных в соответствии с требованиями предметной области; навыками оценки и контроля качества систем и программных средств.</p>

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
	ПК-3	способностью критически переосмысливать накопленный опыт, изменять при необходимости вид и характер своей профессиональной деятельности	разнообразие направлений развития своего профессионализма и мастерства; перспективы использования приобретенных компетенций в различных отраслях производства и научной деятельности	ориентироваться на рынке спроса трудовых услуг по приобретенной профессии; пользоваться различными источниками для получения новых знаний и умений в профессиональной деятельности.	навыками самообразования и повышения мастерства в профессиональной сфере.
	ПК-7	способностью к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	современный уровень развития алгоритмических и программных решений в области системного и прикладного программного обеспечения. элементы проектирования сверхбольших интегральных схем, моделирования и разработки математического обеспечения оптических или квантовых элементов для компьютеров нового поколения, языки программирования, алгоритмы, библиотеки и	разрабатывать программное обеспечение автоматизированных систем вычислительных комплексов, сервисов, операционных систем и распределенных баз данных, разрабатывать языки программирования, алгоритмы, библиотеки и пакеты программ, продукты системного и прикладного программного обеспечения использовать современные программные средства для создания программных продуктов.	навыками применения наукоемких технологий и пакетов программ для решения прикладных задач в области физики, химии, биологии, экономики, медицины, экологии, навыками разработки алгоритмических и программных решений системного и прикладного программного обеспечения.

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
			пакеты программ, продукты системного и прикладного программного обеспечения использование инструментальных средств, автоматизированных систем в научной и практической деятельности		

## 2. Структура и содержание дисциплины

### 2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 2 зач.ед. (72 часов), их распределение по видам работ представлено в таблице (для студентов ОФО).

Вид учебной работы	Всего часов	Семестры (часы)
		4
<b>Контактная работа, в том числе:</b>	52,2	52,2
<b>Аудиторные занятия (всего):</b>	<b>50</b>	<b>50</b>
Занятия лекционного типа	16	16
Лабораторные занятия	34	34
Занятия семинарского типа (семинары, практические занятия)		
<b>Иная контактная работа:</b>	2,2	2,2
Контроль самостоятельной работы (КСР)	2	2
Промежуточная аттестация (ИКР)	0,2	0,2
<b>Самостоятельная работа, в том числе:</b>	19,8	19,8
Курсовая работа		
Проработка учебного (теоретического) материала	10	10
Выполнение индивидуальных заданий	9,8	9,8
Реферат		
Подготовка к текущему контролю		
<b>Контроль: зачет</b>		
Подготовка к экзамену		
<b>Общая трудоемкость</b>	<b>час.</b>	<b>72</b>
	<b>в том числе контактная работа</b>	<b>52,2</b>
	<b>зач. ед</b>	<b>2</b>

## 2.2 Структура дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы дисциплины, изучаемые в 4 семестре (для студентов ОФО)

1	Наименование разделов	Количество часов						
		Всего	Контактная работа				Контр оль	Самост оательн ая работа
			Л	ЛР	КСР	ИК Р		
2	3	4	5	6	7	8	9	
1	Типы данных и операторы языка C#	15,8	4	8				3,8
2	Функции. Ввод - вывод	18	4	8				6
3	Объектно-ориентированное программирование	18	4	8	2			4
4	Конструкторы и деструкторы Наследование и перегрузка	20	4	10				6
	<b>Итого по дисциплине:</b>	71,8	16	34	2			19,8
	Промежуточная аттестация (ИКР)	0,2				0,2		
	<i>Контроль</i>							
	<i>Всего:</i>	72	16	34	2	0,2		19,8

## 2.3 Содержание разделов дисциплины:

**Раздел 1.** Основные типы данных. Константы. Выражения и операторы C#. Операции логического сдвига. Условные операторы. Операторы циклов. Указатели и ссылки. Разыменование указателей. Адресная арифметика. Одномерные массивы. Многомерные массивы.

**Раздел 2.** Функции в C#. Область действия имен. Способы передачи параметров в функции. Параметры командной строки. Структуры. Массивы структур. Указатели на структуры. Массивы структур с использованием указателей. Объединения. Директивы препроцессора. Макросы. Классы памяти. Функции ввода-вывода символов. Функции ввода-вывода строк. Форматированный ввод-вывод. Файловый ввод-вывод. Функции ввода-вывода символов. Файловый ввод-вывод. Функции ввода-вывода строк. Файловый форматированный ввод-вывод. Неформатированный ввод-вывод в нестандартные файловые потоки.

**Раздел 3.** Определение классов в C#. Свойства и методы. Работа с объектами через указатели. Неполное объявление класса.

**Раздел 4.** Конструкторы и деструкторы. Конструкторы с параметрами. Инициализация. Конструкторы по умолчанию. Статические данные и функции. Дружественные функции и классы. Наследование. Перегрузка функций. Перегрузка операций.

### 2.3.1 Занятия лекционного типа

№	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
1	Типы данных и операторы языка C#	Основные типы данных. Константы. Выражения и операторы C#. Операции логического сдвига. Условные операторы. Операторы циклов. Указатели и ссылки. Разыменование указателей. Адресная арифметика. Одномерные массивы. Многомерные массивы.	Вопросы для устного опроса
2	Функции. Ввод - вывод	Функции в C#. Область действия имен. Способы передачи параметров в функции. Параметры командной строки. Структуры. Массивы структур. Указатели на структуры. Массивы структур с использованием указателей. Объединения. Директивы препроцессора. Макросы. Классы памяти. Функции ввода-вывода символов. Функции ввода-вывода строк. Форматированный ввод-вывод. Файловый ввод-вывод. Функции ввода-вывода символов. Файловый ввод-вывод. Функции ввода-вывода строк. Файловый форматированный ввод-вывод. Неформатированный ввод-вывод в нестандартные файловые потоки.	Вопросы для устного опроса
3	Объектно-ориентированное программирование	Определение классов в C#. Свойства и методы. Работа с объектами через указатели.	Вопросы для устного опроса

		Неполное объявление класса.	
4	Конструкторы и деструкторы Наследование и перегрузка	Конструкторы и деструкторы. Конструкторы с параметрами. Инициализация. Конструкторы по умолчанию. Статические данные и функции. Дружественные функции и классы. Наследование. Перегрузка функций. Перегрузка операций.	Вопросы для устного опроса

### Вопросы для устного опроса

1. Основные типы данных. Константы.
2. Выражения и операторы C#. Операции логического сдвига.
3. Условные операторы.
4. Операторы циклов.
5. Указатели и ссылки. Разыменование указателей.
6. Адресная арифметика.
7. Одномерные массивы.
8. Многомерные массивы.
9. Функции в C#.
10. Область действия имен.
11. Способы передачи параметров в функции.
12. Параметры командной строки.
13. Структуры. Массивы структур.
14. Указатели на структуры. Массивы структур с использованием указателей.
15. Объединения.
16. Макросы.
17. Функции ввода-вывода строк.
18. Форматированный ввод-вывод.
19. Файловый ввод-вывод. Функции ввода-вывода строк.
20. Файловый форматированный ввод-вывод.
21. Неформатированный ввод-вывод в нестандартные файловые потоки.
22. Определение классов в C#. Свойства и методы.
23. Работа с объектами через указатели.
24. Конструкторы и деструкторы.
25. Конструкторы с параметрами. Инициализация.
26. Конструкторы по умолчанию.
27. Статические данные и функции.
28. Массивы объектов.
29. Контейнеры. Векторы. Итераторы. Алгоритмы.
30. Дружественные функции и классы.
31. Наследование.

32. Перегрузка функций

33. Перегрузка операций.

### 2.3.2 Занятия семинарского типа

Семинарские занятия - не предусмотрены

### 2.3.3 Лабораторные занятия

№	Наименование раздела	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1	Типы данных и операторы языка C#	Основные типы данных. Константы. Выражения и операторы C#. Операции логического сдвига. Условные операторы. Операторы циклов. Указатели и ссылки. Разыменование указателей. Адресная арифметика. Одномерные массивы. Многомерные массивы.	лабораторная
2	Функции. Ввод - вывод	Функции в C#. Область действия имен. Способы передачи параметров в функции. Параметры командной строки. Структуры. Массивы структур. Указатели на структуры. Массивы структур с использованием указателей. Объединения. Директивы препроцессора. Макросы. Классы памяти. Функции ввода-вывода символов. Функции ввода-вывода строк. Форматированный ввод-вывод. Файловый ввод-вывод. Функции ввода-вывода символов.	лабораторная

		Файловый ввод-вывод. Функции ввода-вывода строки. Файловый форматированный ввод- вывод. Неформатированный ввод-вывод в нестандартные файловые потоки.	
3	Объектно- ориентированное программирование	Определение классов в C#. Свойства и методы. Работа с объектами через указатели. Неполное объявление класса.	лабораторная
4	Конструкторы и деструкторы Наследование и перегрузка	Конструкторы и деструкторы. Конструкторы с параметрами. Инициализация. Конструкторы по умолчанию. Статические данные и функции. Дружественные функции и классы. Наследование. Перегрузка функций. Перегрузка операций.	лабораторная

### 2.3.4 Примерная тематика курсовых работ (проектов)

Курсовые работы не предусмотрены

## 2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

№	Наименование раздела	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1.	Проработка учебного (теоретического) материала	Биллинг В.А. Основы объектного программирования на C#: учебное пособие/В.А. Биллинг.-М.:Интернет- Университет Информационных Технологий,2013.- 582с. Павловская Т.А. Программирование на языке C++,

		Курс лекций, М., Интуит НОУ, 2016, 155 стр., [Электронный ресурс], <a href="http://www.book.ru/book/918123">http://www.book.ru/book/918123</a> .
2.	Выполнение индивидуальных заданий	Эйдлина Г.М., Милорадов К.А. Delphi: программирование в примерах и задачах: практикум: учебное пособие/Г.М. Эйдлина, К.А. Милорадов. - М.:РИОР: Инфра-М,2012.-113с. Павловская Т.А. Программирование на языке С++, Курс лекций, М., Интуит НОУ, 2016, 155 стр., [Электронный ресурс], <a href="http://www.book.ru/book/918123">http://www.book.ru/book/918123</a> .

### 3. Образовательные технологии

Удельный вес занятий, проводимых в интерактивных формах, определяется главной целью программы, особенностью контингента обучающихся и содержанием конкретных дисциплин.

Сочетание видов ОД с различными методами ее активизации.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	Количество часов
4	ЛК	Использование средств мультимедиа (компьютерные классы).	4
	ЛР	Обучение на основе опыта.	2
<b>Итого</b>			<b>6</b>

В соответствии с требованиями ФГОС ВО по направлению подготовки бакалавров программа по дисциплине «Программирование на С#» предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; разбор конкретных ситуаций.

Компьютерные технологии позволяют проводить сравнительный анализ научных исследований по данной проблеме, являясь средством разнопланового отображения алгоритмов и демонстрационного материала.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и бакалаврами во время лекций и анализа результатов самостоятельной работы. Это обусловлено тем, что в процессе работы с языком программирования С# часто встречаются задачи, для которых единых подходов не существует. Каждая конкретная задача при своем исследовании имеет множество подходов, а это требует разбора и оценки целой совокупности конкретных ситуаций. Особенно этот подход широко используется при определении адекватности математической модели, результатам компьютерных экспериментов.

Цель *лекции* – обзор понятий программирования на С#.

Цель *лабораторного занятия* – научить применять теоретические знания при решении и исследовании конкретных задач.

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

#### **4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации**

##### **4.1 Фонд оценочных средств для проведения текущей аттестации**

Учебная деятельность проходит в соответствии с графиком учебного процесса. Процесс самостоятельной работы контролируется во время аудиторных занятий и индивидуальных консультаций. Самостоятельная работа студентов проводится в форме изучения отдельных теоретических вопросов по предлагаемой литературе и решения индивидуальных задач повышенной сложности.

Фонд оценочных средств дисциплины состоит из средств текущего контроля (см. примерные варианты контрольных работ, индивидуальных заданий, задач и вопросов) и итоговой аттестации (зачета).

В качестве оценочных средств, используемых для текущего контроля успеваемости, предлагается перечень вопросов, которые прорабатываются в процессе освоения курса. Данный перечень охватывает все основные разделы курса, включая знания, получаемые во время самостоятельной работы.

Оценка успеваемости осуществляется по результатам: самостоятельного выполнения лабораторных работ, устного опроса при сдаче выполненных самостоятельных заданий, ответов на зачете.

Аттестация по учебной дисциплине проводится в виде зачета. Зачетный билет содержит два теоретических вопроса и одну задачу. Студент готовит ответы на билет в письменной форме в течение установленного времени. Далее зачет протекает в форме собеседования.

#### **Примерное содержание лабораторных работ**

1. Создать массив типа *int* на 10 элементов и заполнить его случайными числами от 7 до 14. После заполнения перезаписать все числа, которые больше десяти: от хранимого значение отнять 10. Например в ячейке хранится число 12:  $12 - 10 = 2$ . Записать в эту ячейку 2. Записывая новое значение, используйте составные (комбинированные) операторы.

2. Заполнить массив из 50-ти элементов нечётными числами от 1 до 99. (используйте операцию остаток от деления, чтобы проверить число на чётность)

3. Объявить три массива. Первые два заполнить случайными значениями от 10 до 30. В элементы третьего массива записать сумму соответствующих элементов первых двух массивов. (в нулевую ячейку третьего – сумму нулевых ячеек первого и второго массивов и так далее). Затем найти среднее арифметическое элементов третьего массива, максимальное значение и минимальное значение, которые он хранит.

4. Объявить два целочисленных массива с разными размерами и написать функцию, которая заполняет их элементы значениями и показывает на экран. Функция должна принимать два параметра – массив и его размер.

5. Необходимо создать двумерный массив 5 x 5. Далее написать функцию, которая заполнит его случайными числами от 30 до 60. Создать еще две функции, которые находят максимальный и минимальный элементы этого двумерного массива.

6. Написать игру в которой имитируется бросание кубиков компьютером и пользователем. В игре 2 кубика и на каждом из них может выпасть от 1 до 6 очков. Реализовать определение программой первого ходящего. Каждый делает по четыре броска. После бросков показать, нарисованные символами кубики и количество очков, выпавших на них. После пары бросков (бросок компьютера + бросок пользователя) выводить на экран промежуточный результат – количество набранных очков игроком и компьютером. В конце сообщить о том, кто выиграл по итогам всех бросков.

7. Создать класс *Tiles* (кафель), который будет содержать поля с открытым доступом: *brand*, *size\_h*, *size\_w*, *price* и метод класса *getData()*. В главной функции объявить пару объектов класса и внести данные в поля. Затем отобразить их, вызвав метод *getData()*.

8. Определить класс *Children*, который содержит такие поля (члены класса): закрытые – имя ребенка, фамилию и возраст, публичные – методы ввода данных и отображения их на экран. Объявить два объекта класса, внести данные и показать их.

9. Преобразовать строки двумерного массива в столбцы. Например было:

<b>11</b>	<b>22</b>
<b>33</b>	<b>44</b>
<b>55</b>	<b>66</b>
<b>77</b>	<b>88</b>

стало:

<b>11</b>	<b>33</b>	<b>55</b>	<b>77</b>
<b>22</b>	<b>44</b>	<b>66</b>	<b>88</b>

Класс будет содержать два метода: один заполнит массив значениями, второй произведет замену значений строк на значения столбцов.

10. Объявите указатель на массив типа *double* и предложите пользователю выбрать его размер. Далее напишите четыре функции: первая должна выделить память для массива, вторая – заполнить ячейки данными, третья – показать данные на экран, четвертая – освободить занимаемую память. Программа должна предлагать пользователю продолжать работу (создавать новые динамические массивы) или выйти из программы.

11. Объявите указатель на массив типа `int` и выделите память для 12-ти элементов. Необходимо написать функцию, которая поменяет значения четных и нечетных ячеек массива. Например, есть массив из 4-х элементов:

	ячейка 0	ячейка 1	ячейка 2	ячейка 3
Исходные данные массива	1	2	3	4

  

	ячейка 0	ячейка 1	ячейка 2	ячейка 3
Данные после работы функции	2	1	4	3

12. Объявить и заполнить двумерный динамический массив случайными числами от 10 до 50. Показать его на экран. Для заполнения и показа на экран написать отдельные функции. (подсказка: функции должны принимать три параметра – указатель на динамический массив, количество строк, количество столбцов). Количество строк и столбцов выбирает пользователь.

### Примерное содержание самостоятельных работ

Самостоятельная работа студента заключается в самостоятельном выборе темы проекта-программы, его архитектуры и полное создание его с основными функциональными возможностями.

### Вопросы теста по языку C#.

Тест 1.

<http://chipenable.ru/index.php/programming-c/68-c-test-the-0x10-best-questions-for-would-be-embedded-programmers.html>

1. Используя директиву `#define`, как бы вы описали именованную константу, которая возвращает число секунд в году? Високосными годами следует пренебречь.

```
#define SECONDS_PER_YEAR (60UL * 60UL * 24UL * 365UL)
```

2. Напишите «стандартный» макрос `MIN`. То есть, макрос, который берет два аргумента и возвращает меньший из них.

```
#define MIN(A,B) ((A) <= (B) ? (A) : (B))
```

3. Каково назначение директивы препроцессора `#error`?

4. При программировании встраиваемых систем часто используются бесконечные циклы. Как реализовать бесконечный цикл в Си?

5. Используя переменную «a», запишите объявления для:

- (a) Целого;
- (b) Указателя на целое;
- (c) Указателя на указатель на целое;
- (d) Массива из десяти целых;
- (e) Массива из десяти указателей на целые;
- (f) Указателя на массив из десяти целых;
- (g) Указателя на функцию, которая принимает целочисленный аргумент и возвращает целое;
- (h) Массива из десяти указателей на функции, которая принимает целочисленный аргумент и возвращает целое;

6. В каких случаях используется ключевое слово `static`?

7. Что означает ключевое слово `const`?

8. Что означает ключевое слово `volatile`? Приведите три различных примера его использования.

9. При программировании встраиваемых систем приходится часто манипулировать битами в регистрах или переменных. Дана целая переменная «a», напишите два фрагмента кода. Первый должен установить 3-ий бит этой переменной. Второй должен очищать его. В обоих случаях, другие биты должны остаться без изменений.

10. Прерывания являются важной частью встраиваемых систем. Поэтому многие производители компиляторов предлагают к стандартному языку Си расширение для поддержки прерываний. Обычно - это новое ключевое слово `__interrupt`. Следующий код использует `__interrupt`, чтобы описать программу обработки прерываний. Прокомментируйте его.

11. Что делает следующий фрагмент кода и почему?

```
void foo(void)
{
    unsigned int a = 6;
    int b = -20;
    (a+b > 6) ? puts("> 6?") : puts("<= 6?");
}
```

12. Прокомментируйте следующий фрагмент кода:

```
unsigned int zero = 0;
```

```
unsigned int compzero = 0xFFFF; /* 1's complement of zero */
```

13. Что делает следующий фрагмент кода и почему?

```
char *ptr;  
if ((ptr = (char *)malloc(0)) == NULL) {  
    puts("Got a null pointer");  
}  
else {  
    puts("Got a valid pointer");  
}
```

14. Си позволяет некоторые ужасные конструкции. Допустима ли эта конструкция, и если да, то что делает этот код?

```
int a = 5, b = 7, c;
```

```
c = a+++b;
```

Тест 2.

Зарезервированными словами в языке программирования C#, являются:

- a.  fortran
- b.  paskal
- c.  integral

ключевыми словами в языке программирования C#, являются:

- a.  extern
- b.  back
- c.  return

Формат константы с плавающей точкой в языке программирования C#, имеет вид: [ цифры ].[индекс] [ E|e [+|-] цифры ]

- a.  Неправильно
- b.  Правильно

Тип int в языке программирования C#, имеет следующий диапазон:

- a.  0X8000 - 0XFFFF
- b.  0 - 077777

функции в языке программирования C#, не могут возвращать:

- a.  массивы
- b.  функции

Ключевые слова для определения Целых типов данных в языке программирования C#, являются:

- a.  unsigned
- b.  signed
- c.  short
- d.  long double
- e.  long

Ключевые слова для определения плавающих типов данных в языке программирования C#, являются:

- a.  float
- b.  double
- c.  extra double
- d.  long float
- e.  long double

Величина типа double в языке программирования C#, занимает:

- a.  8 бит в памяти
- b.  4 бита в памяти
- c.  4 байта в памяти
- d.  24 бита в памяти
- e.  2 бита в памяти

Величина с модификатором типа float языка программирования C#, занимает:

- a.  4 байта в памяти
- b.  8 байт в памяти
- c.  8 бит в памяти
- d.  16 бит в памяти
- e.  2 байта в памяти

В структуре struct языка программирования C#, обязательно должен быть указан:

- a.  1 индекс
- b.  1 указатель
- c.  1 тип
- d.  ничего
- e.  несколько компонентов

Элементом структуры языка программирования C#, может быть:

- a.  байтовое поле

- b.  адрес байтового поля
- c.  битовое поле
- d.  адрес битового поля
- e.  Поле нулевой длины

#### **4.2 Фонд оценочных средств для проведения промежуточной аттестации**

Перечень вопросов к зачету

1. Основные типы данных. Константы.
2. Выражения и операторы C#. Операции логического сдвига.
3. Условные операторы.
4. Операторы циклов.
5. Указатели и ссылки. Разыменование указателей.
6. Адресная арифметика.
7. Одномерные массивы.
8. Многомерные массивы.
9. Функции в C#.
10. Область действия имен.
11. Способы передачи параметров в функции.
12. Параметры командной строки.
13. Структуры. Массивы структур.
14. Указатели на структуры. Массивы структур с использованием указателей.
15. Объединения.
16. Макросы.
17. Функции ввода-вывода строк.
18. Форматированный ввод-вывод.
19. Файловый ввод-вывод. Функции ввода-вывода строк.
20. Файловый форматированный ввод-вывод.
21. Неформатированный ввод-вывод в нестандартные файловые потоки.
22. Определение классов в C#. Свойства и методы.
23. Работа с объектами через указатели.
24. Конструкторы и деструкторы.
25. Конструкторы с параметрами. Инициализация.
26. Конструкторы по умолчанию.
27. Статические данные и функции.
28. Массивы объектов.
29. Контейнеры. Векторы. Итераторы. Алгоритмы.
30. Дружественные функции и классы.
31. Наследование.
32. Перегрузка функций
33. Перегрузка операций.

#### **5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

### 5.1 Основная литература:

1. Павловская, Татьяна Александровна. С/С++. Программирование на языке высокого уровня [Текст] : для магистров и бакалавров : учебник для студентов вузов, обучающихся по направлению подготовки дипломированных специалистов "Информатика и вычислительная техника" / Т. А. Павловская. - СПб. [и др.] : ПИТЕР, 2013. - 460 с.
2. Биллиг, В.А. Основы программирования на С# 3.0: ядро языка [Электронный ресурс]/ В.А. Биллиг. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 411 с. : ил. - URL: <http://biblioclub.ru/index.php?page=book&id=428947>

### 5.2 Дополнительная литература:

1. Царев, Р.Ю. Программирование на языке Си [Электронный ресурс] : учебное пособие / Р.Ю. Царев ; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск : Сибирский федеральный университет, 2014. - 108 с. : табл., схем. - URL: <http://biblioclub.ru/index.php?page=book&id=364601>
2. Языки программирования [Электронный ресурс]: лабораторный практикум / Министерство образования и науки РФ, Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет» ; сост. Е.А. Малиновская, Р.А. Рыскаленко. - Ставрополь : СКФУ, 2016. - Ч. 1. - 103 с. : ил. - URL: <http://biblioclub.ru/index.php?page=book&id=467412>
3. Туральчук, К.А. Параллельное программирование с помощью языка С# [Электронный ресурс]/ К.А. Туральчук. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 190 с. : ил. - URL: <http://biblioclub.ru/index.php?page=book&id=429098>
4. Павловская, Татьяна Александровна. С/С++. Программирование на языке высокого уровня [Текст] : учебник для студентов вузов / Т. А. Павловская. - СПб. [и др.] : ПИТЕР, 2006. - 460 с

### 5.3. Периодические издания:

1. Вестник МГУ сер.1 Математика. Механика.

## 6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

№	Наименование электронного ресурса	Ссылка на электронный адрес
1.	Электронный каталог Научной библиотеки КубГУ	<a href="https://www.kubsu.ru/">https://www.kubsu.ru/</a>
2.	Электронная библиотечная система «BOOK.ru» ООО «КноРус медиа»	<a href="https://www.book.ru">https://www.book.ru</a>
3.	Электронная библиотечная система "Университетская библиотека ONLINE"	<a href="http://www.biblioclub.ru">www.biblioclub.ru</a>
4.	Электронная библиотечная система «ZNANIUM.COM» ООО «ЗНАНИУМ»	<a href="http://www.znanium.com">www.znanium.com</a>

5.	Электронная библиотечная система издательства "Лань"	<a href="http://e.lanbook.com/">http://e.lanbook.com/</a>
6.	Электронная библиотечная система "Юрайт"	<a href="http://www.biblio-online.ru">http://www.biblio-online.ru</a>

1. <http://docs.oracle.com/javase/tutorial/> Сайт Oracle
2. Единая коллекция цифровых образовательных ресурсов. URL: <http://school-collection.edu.ru/>
3. Информационная система «Единое окно доступа к образовательным ресурсам». URL: <http://window.edu.ru/>
4. Российское образование. Федеральный портал. URL: <http://www.edu.ru/>  
Сайт Министерства образования и науки Российской Федерации <http://минобрнауки.рф/>
5. Университетская библиотека ONLINE URL: <http://www.biblioclub.ru/>
6. Федеральный портал «Российское образование» URL: <http://www.edu.ru/>
7. Федеральный центр информационно-образовательных ресурсов. URL: <http://fcior.edu.ru/>
8. Электронная библиотека “Социология, психология, управление” URL: <http://soc.lib.ru>
9. Электронная библиотечная система издательства "Лань". URL: <http://e.lanbook.com/>
10. Электронный каталог Научной библиотеки КубГУ. URL: <http://www.kubsu.ru/University/library/resources/Poisk2012.php>

## **7. Методические указания для обучающихся по освоению дисциплины**

При изучении курса «Программирование на C#» необходимо активизировать остаточные знания студентов по таким дисциплинам как системное программное обеспечение, языки программирования и методы трансляции, программирование в ОС MS Windows.

При чтении лекционного курса представляется целесообразным обратить внимание на физические приложения излагаемых математических фактов. Чтобы изложение было понятным, следует акцентировать внимание не столько на формальных моментах доказательств, сколько на движущих ими идеях.

Необходимо отметить практическую значимость соответствующих проблем, обратить внимание на требования, предъявляемые к современному специалисту – прикладному математику, пояснить необходимость использования полученных знаний при изучении последующих специальных курсов.

## **8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (при необходимости)**

### **8.1 Перечень необходимого программного обеспечения**

Операционная система - Microsoft Windows, Архиватор WinRAR, Браузер Internet Explorer, Google Chrome, Mozilla Firefox. Пакет программ Microsoft Office. Графические пакеты - Adobe Design Standard CS3, CorelDRAW Graphics Suite X3. Математический пакет программ - CodeGear RAD Studio.

## 8.2 Перечень необходимых информационных справочных систем

Информационных справочных систем по этому предмету не предусмотрено.

## 9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

№	Наименование специальных помещений и помещений для самостоятельной работы	Номера аудиторий / кабинетов
1.	учебные аудитории для проведения занятий лекционного типа	501,502,503,505,506,507,508, 509, 510,513,514
2.	учебные аудитории для проведения занятий семинарского типа	501,502,503,505,506,507,508, 509, 510,513,514
3.	Компьютерные классы с выходом в Интернет	503,509,510
4.	учебные аудитории для выполнения научно – исследовательской работы (курсового проектирования)	Кабинет курсового проектирования (выполнения курсовых работ) - № 503 Оборудование: мультимедийный проектор, экран, персональные компьютеры, учебная мебель, доска учебная, выход в Интернет, учебно-наглядные пособия (тематические иллюстрации), принтер, презентации на электронном носителе, сплит-система
5.	учебные аудитории для самостоятельной работы, с рабочими местами, оснащенными компьютерной техникой с подключением к сети «Интернет» и обеспечением неограниченного доступа в электронную информационно-образовательную среду организации для каждого обучающегося, в соответствии с объемом изучаемых дисциплин	Кабинет для самостоятельной работы - № 504 Оборудование: персональные компьютеры, учебная мебель, доска учебная, выход в Интернет
6.	Исследовательские лаборатории (центров), оснащенные лабораторным оборудованием	Компьютерный класс № 510 : мультимедийный проектор, экран, персональные компьютеры, учебная мебель, доска учебная, выход в Интернет, наглядные пособия. Сетевое оборудование CISCO (маршрутизаторы, коммутаторы, 19-ти дюймовый сетевой шкаф) сплит-система, стенд «Архитектура ПЭВМ»
7.	Кабинет групповых и индивидуальных	№508

	консультаций	Оборудование: персональный компьютер, учебная мебель, доска учебная, учебно-наглядные пособия (тематические иллюстрации), сканер, доска магнитно-маркерная, стеллажи с учебной и периодической литературой
8.	Помещение для хранения и профилактического обслуживания учебного оборудования	Помещение № 511, Помещение № 516, Помещение № 517, Помещение № 518
9.	Помещение для проведения текущей и промежуточной аттестации	501,502,503,505,506,507,508, 509, 510,513,514

Согласно письма Министерства образования и науки РФ № МОН-25486 от 21.06.2017г «О разработке адаптированных образовательных программ» -Разработка адаптивной программы необходима в случае наличия в образовательной организации хотя бы одного обучающегося с ограниченными возможностями здоровья

Для обучающихся из числа инвалидов обучение проводится организацией с учетом особенностей их психофизического развития, их индивидуальных возможностей и состояния здоровья (далее - индивидуальные особенности).

При проведении обучения инвалидов обеспечивается соблюдение следующих общих требований:

- проведение обучения для инвалидов в одной аудитории совместно с обучающимися, не имеющими ограниченных возможностей здоровья, если это не создает трудностей для обучающихся;

- присутствие в аудитории ассистента (ассистентов), оказывающего обучающимся инвалидам необходимую техническую помощь с учетом их индивидуальных особенностей;

- пользование необходимыми обучающимся инвалидам техническими средствами с учетом их индивидуальных особенностей;

- обеспечение возможности беспрепятственного доступа обучающихся инвалидов в аудитории, туалетные и другие помещения, а также их пребывания в указанных помещениях;

В зависимости от индивидуальных особенностей обучающихся с ограниченными возможностями здоровья, организация обеспечивает выполнение следующих требований при проведении занятий:

*а) для слепых:*

- на компьютере со специализированным программным обеспечением для слепых, либо надиктовываются ассистенту;

*б) для слабовидящих:*

- задания и иные материалы оформляются увеличенным шрифтом;

- обеспечивается индивидуальное равномерное освещение не менее 300

люкс;

-при необходимости обучающимся предоставляется увеличивающее устройство, допускается использование увеличивающих устройств, имеющихся у обучающихся;

в) для глухих и слабослышащих, с тяжелыми нарушениями речи:

-обеспечивается наличие звукоусиливающей аппаратуры коллективного пользования, при необходимости обучающимся предоставляется звукоусиливающая аппаратура индивидуального пользования;

г) для лиц с нарушениями опорно-двигательного аппарата (тяжелыми нарушениями двигательных функций верхних конечностей или отсутствием верхних конечностей):

-письменные задания выполняются обучающимися на компьютере со специализированным программным обеспечением или надиктовываются ассистенту;

Обучающийся инвалид при поступлении подает письменное заявление о необходимости создания для него специальных условий при проведении обучения с указанием особенностей его психофизического развития, индивидуальных возможностей и состояния здоровья (далее - индивидуальные особенности). К заявлению прилагаются документы, подтверждающие наличие у обучающегося индивидуальных особенностей (при отсутствии указанных документов в организации).