

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Чувашский государственный университет имени И. Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра компьютерных технологий

«УТВЕРЖДАЮ»  
Проректор по учебной работе

  
И.Е. Поверин

«31» августа 2017 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
**«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»**

Направление подготовки (специальность) 09.03.04 Программная инженерия

Квалификация (степень) выпускника – бакалавр

Профиль (направленность) Управление разработкой программных проектов

Прикладной бакалавриат

Чебоксары – 2017

Рабочая программа основана на требованиях Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 «Программная инженерия», утвержденного приказом Министерства образования и науки Российской Федерации № 229 от 12.03.2015 г.

*СОСТАВИТЕЛЬ (СОСТАВИТЕЛИ):*

доцент, к.т.н.

 П.В. Желтов

старший преподаватель

 Е.А. Мытникова

*ОБСУЖДЕНО:*

на заседании кафедры компьютерных технологий « 30 » 08 2017г., протокол № 1

заведующий кафедрой

 Т.А. Лавина

*СОГЛАСОВАНО:*

Методическая комиссия факультета информатики и вычислительной техники  
« 30 » 08 2017г., протокол № 1

Декан факультета

 А.В. Щипцова

Директор научной библиотеки

 Н. Д. Никитина

Начальник управления информатизации

 И. П. Пивоваров

Начальник учебно-методического управления

 В. И. Маколов

## Оглавление

<b>1. Цель и задачи обучения по дисциплине</b> .....	4
<b>2. Место дисциплины в структуре основной образовательной программы (ООП)</b> .....	4
<b>3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ООП</b> .....	4
<b>4. Структура и содержание дисциплины</b> .....	4
4.1. Содержание дисциплины .....	5
4.2. Объем дисциплины, виды учебной работы обучающихся по очной форме обучения .....	7
<b>5. Содержание разделов дисциплины</b> .....	8
5.1. Лекции .....	8
5.2. Лабораторные работы .....	12
5.3. Вопросы для самостоятельной работы студента в соответствии с содержанием разделов дисциплины .....	13
<b>6. Образовательные технологии</b> .....	13
<b>7. Формы аттестации и оценочные материалы</b> .....	14
7.1. Вопросы к зачету .....	14
7.2. Вопросы и задачи к экзамену .....	15
7.3. Выполнение и примерная тематика курсового проекта.....	17
7.4. Выполнение и примерные задания расчетно-графической работы .....	18
7.5. Выполнение и примерная тематика контрольной работы .....	18
<b>8. Учебно-методическое и информационное обеспечение дисциплины</b> .....	18
8.1. Рекомендуемая основная литература (ежегодное обновление перечня и условия доступа представлены в Приложениях к рабочей программе) .....	18
8.2. Рекомендуемая дополнительная литература (ежегодное обновление и условия доступа перечня представлены в Приложениях к рабочей программе).....	18
8.3. Рекомендуемые методические разработки по дисциплине .....	18
8.4. Программное обеспечение, профессиональные базы данных, информационно-справочные системы. 19	
8.5. Рекомендуемые интернет-ресурсы и открытые он-лайн курсы .....	19
<b>9. Материально-техническое обеспечение дисциплины</b> .....	19
<b>10. Средства адаптации преподавания дисциплины к потребностям лиц с ограниченными возможностями</b> .....	19
<b>11. Методические рекомендации по освоению дисциплины</b> .....	20

## 1. Цель и задачи обучения по дисциплине

Целью изучения дисциплины является обучение студентов методике разработки программных средств с использованием технологии объектно-ориентированного программирования (ООП) на языке C#.

Задачи изучения дисциплины:

- освоение и применение средств автоматизированного проектирования, разработки, тестирования и сопровождения программного обеспечения;
- участие в процессах разработки программного обеспечения.

## 2. Место дисциплины в структуре основной образовательной программы (ООП)

Блок дисциплин – Б1.В Дисциплины (модули). Вариативная часть

Изучение данной дисциплины базируется на курсах «Информатика и программирование», «Программирование на языках высокого уровня», «Введение в профессиональную деятельность программиста».

Последующие дисциплины, для успешного изучения которых необходима данная дисциплина: «Проектный практикум», «Проектирование и конструирование программного обеспечения» «Разработка мобильных приложений», «Численные методы в задачах обработки данных».

## 3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ООП

Процесс обучения по дисциплине направлен на формирование следующих компетенций:

профессиональных (ПК), соответствующими виду (видам) профессиональной деятельности, на которой (которые) ориентирована ООП:

- владением навыками использования различных технологий разработки программного обеспечения (ПК-3);

В результате обучения по дисциплине, обучающийся должен (ЗУН):

знать:

- 31 - концепции объектно-ориентированного программирования;
- 32 – основные понятия объектно-ориентированного программирования;
- 33 – основы графики Windows Forms;
- 34 – коллекции и итераторы;
- 35 – дополнительные возможности языка C#;

уметь:

- У1 – описывать классы;
- У2 – строить отношения между классами;
- У3 – пользоваться XAML;
- У4 – пользоваться языком интегрированных запросов LINQ;

владеть навыками:

- Н1 - программирования на основе платформы .Net Framework;
- Н2 – проектирования визуальных приложений Windows Forms;
- Н3 - обобщенного программирования;
- Н4 - асинхронного программирования.

## 4. Структура и содержание дисциплины

Образовательная деятельность по дисциплине проводится:

- в форме контактной работы обучающихся с педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях (далее – контактная работа);
- в форме самостоятельной работы.

Контактная работа включает в себя занятия лекционного типа, занятия семинарского типа (лабораторные работы), групповые и (или) индивидуальные консультации, в том числе в электронной информационно-образовательной среде.

Обозначения:

Л – лекции, л/р – лабораторные работы, п/р – практические занятия, КСР – контроль самостоятельной работы, СРС – самостоятельная работа студента, ИФР – интерактивная форма работы, К – контроль.

#### 4.1. Содержание дисциплины

Содержание	Формируемые компетенции	Формируемые ЗУН
Раздел 1. Основы языка C#	ПК-3	<b>знать:</b> – 31 - концепции объектно-ориентированного программирования; – 32 – основные понятия объектно-ориентированного программирования; <b>уметь:</b> – У1 – описывать классы; <b>владеть навыками:</b> – Н1 - программирования на основе платформы .Net Framework;
1.1. Введение. Императивная и декларативная парадигмы программирования. Эволюция языков программирования		
1.2. Принципы программирования на основе платформы .Net Framework		
1.3. Основные элементы программ (данные, выражения, операторы)		
1.4. Массивы и строки		
1.5. Методы как основа процедурного программирования		
1.6. Класс как контейнер статических членов		
1.7. Класс как тип. Объекты и их члены		
1.8. Исключения		
1.9. Регулярные выражения		
1.10. Перечисления и структуры – типы значений		
Зачет(1)	ПК-3	<b>знать:</b> – 31 - концепции объектно-ориентированного программирования; – 32 – основные понятия объектно-ориентированного программирования; <b>уметь:</b> – У1 – описывать классы; <b>владеть навыками:</b> - Н1 - программирования на основе платформы .Net Framework;
Раздел 2. Визуальное проектирование приложений Windows Forms	ПК-3	<b>знать:</b> – 33 – основы графики Windows Forms; <b>уметь:</b> – У2 – строить отношения между классами; <b>владеть навыками:</b> – Н2 – проектирования визуальных приложений Windows Forms;
2.1. Визуальное проектирование приложений Windows Forms		
2.2. Основы графики Windows Forms		
2.3. Отношения между классами		
2.4. Делегаты, анонимные методы, лямбда-выражения, события и таймеры		
2.5. Интерфейсы		
Экзамен(1)	ПК-3	<b>знать:</b> – 31 - концепции объектно-

		<p>ориентированного программирования;</p> <ul style="list-style-type: none"> <li>- 32 – основные понятия объектно-ориентированного программирования;</li> <li>- 33 – основы графики Windows Forms;</li> </ul> <p><b>уметь:</b></p> <ul style="list-style-type: none"> <li>- У1 – описывать классы;</li> <li>- У2 – строить отношения между классами;</li> </ul> <p><b>владеть навыками:</b></p> <ul style="list-style-type: none"> <li>- Н1 - программирования на основе платформы .Net Framework;</li> <li>- Н2 – проектирования визуальных приложений Windows Forms;</li> </ul>
Раздел 3. Обобщенное программирование	ПК-3	<p><b>знать:</b></p> <ul style="list-style-type: none"> <li>- 34 – коллекции и итераторы;</li> </ul> <p><b>уметь:</b></p> <ul style="list-style-type: none"> <li>- У3 - пользоваться XAML;</li> </ul> <p><b>владеть навыками:</b></p> <ul style="list-style-type: none"> <li>- Н3 - обобщенного программирования;</li> </ul>
3.1. Обобщенное программирование		
3.2. Коллекции и итераторы		
3.3. Поточковый ввод-вывод		
3.4. XAML и технология WPF		
Зачет(2)		<p><b>знать:</b></p> <ul style="list-style-type: none"> <li>- 34 – коллекции и итераторы;</li> </ul> <p><b>уметь:</b></p> <ul style="list-style-type: none"> <li>- У3 - пользоваться XAML;</li> </ul> <p><b>владеть навыками:</b></p> <ul style="list-style-type: none"> <li>- Н3 - обобщенного программирования;</li> </ul>
Раздел 4. Дополнительные возможности языка C#	ПК-3	<p><b>знать:</b></p> <ul style="list-style-type: none"> <li>- 35 – дополнительные возможности языка C#;</li> </ul> <p><b>уметь:</b></p> <ul style="list-style-type: none"> <li>- У4 - пользоваться языком интегрированных запросов LINQ;</li> </ul> <p><b>владеть навыками:</b></p> <ul style="list-style-type: none"> <li>- Н4 - асинхронного программирования.</li> </ul>
4.1. Язык интегрированных запросов LINQ и элементы функционального программирования		
4.2. Асинхронные методы и параллельные программы		
4.3. Небезопасный код и препроцессорные директивы C#		
4.4. Мобильные приложения и Web-приложения		
Экзамен(2)	ПК-3	<p><b>знать:</b></p> <ul style="list-style-type: none"> <li>- 31 - концепции объектно-ориентированного программирования;</li> <li>- 32 – основные понятия объектно-ориентированного программирования;</li> <li>- 33 – основы графики Windows Forms;</li> <li>- 34 – коллекции и итераторы;</li> <li>- 35 – дополнительные возмож-</li> </ul>



3.1. Обобщенное программирование	10	4	4			10	4	
3.2. Коллекции и итераторы	22	4	4			10	4	
3.3. Поточковый ввод-вывод	22	4	4			10	4	
3.4. XAML и технология WPF	22	4	4			10	4	
Раздел 4. Дополнительные возможности языка C#								
4.1. Язык интегрированных запросов LINQ и элементы функционального программирования	13	2	6			5	6	
4.2. Асинхронные методы и параллельные программы	13	2	6			5	6	
4.3. Небезопасный код и препроцессорные директивы C#	13	2	6			5	6	
4.4. Мобильные приложения и Web-приложения	13	2	6			5	6	
Зачет(1)	10					10		
Экзамен(1)	36							36
Зачет(2)	9					9		
Экзамен(2)	38				2			36
<b>Итого</b>	<b>360</b>	<b>56</b>	<b>88</b>		<b>2</b>	<b>142</b>	<b>88</b>	<b>72</b>
<b>Зачетных единиц</b>	<b>10</b>							

Вид промежуточной аттестации: зачет в 5 семестре, экзамен в 6 семестре, зачет в 7 семестре, экзамен в 8 семестре.

## 5. Содержание разделов дисциплины

### 5.1. Лекции

#### Раздел 1. Основы языка C#

*Тема 1.1. Введение. Императивная и декларативная парадигмы программирования. Эволюция языков программирования*

Задачи, цели, содержание и порядок прохождения дисциплины. Данные и их типы. Тип в математике и в программировании. Парадигмы программирования. Абстракция классов и абстракция функциональная, декларативный и императивный подходы. Алгоритм и программа. Структурный подход к разработке алгоритмов. Процедурно-ориентированное программирование. Языки высокого уровня, поддерживающие процедурно-ориентированный подход. Объектно-ориентированное программирование. Типы, классы, объекты. Языки высокого уровня, поддерживающие объектно-ориентированный подход. Типы приложений.

#### *Тема 1.2. Принципы программирования на основе платформы .Net Framework*

Основные компоненты платформы .Net Framework: общезыковая исполняющая среда (CLR) и библиотека классов (FCL). Управляемый код. Общая система типов (CTS) и общезыковая спецификация (CLS). Типы значений, ссылочные типы и базовые типы в CLR и в языке C#. Виды приложений среды .NET. Интегрированная среда для разработки приложений. Проекты и решения, их создание и применения. Средства навигации (проводник решений). Структура исходного кода консольного приложения. Кодирование, трансляция, синтаксический анализ, отладка и исполнение простой программы.



*Тема 1.3. Основные элементы программ (данные, выражения, операторы)*

Определение класса. Пространство имен. Статические поля и методы. Типы значений и типы ссылок. Базовые типы: предельные значения, свойства и методы. Константы, переменные, именованные константы. Типы C# как классы платформы .NET Framework. Арифметические операции и выражения. Проблема преобразования типов. Явное преобразование типов. Присваивание простое и составное. Проблема преобразования типов при присваивании. Инициализация переменных. Целочисленные арифметические выражения. Переполнения при операциях с целыми. Особые ситуации в арифметических выражениях. Логические операции и операции отношения. Логические выражения. Условная (тернарная) операция. Операции сдвигов. Особенности выполнения операции сдвига применительно к знаковым и беззнаковым типам. Битовые операции. Применение битовых операций для проверки состояния битов и установки битов в заданное значение. Приоритеты операций. Управляющие структуры. Блок операторов. Условные операторы. Вложенность управляющих структур. Множественное ветвление (переключатели). Циклы и средства управления итерациями.

*Тема 1.4. Массивы и строки*

Одномерные массивы: объявление и инициализация. Доступ к элементу массива. Операции с массивом и элементами массива. Основные свойства и методы класса System.Array. Особенности операции присваивания применительно к ссылкам. Проблема потери ссылок. Сборка мусора. Многомерные массивы: объявление и инициализация. Доступ к элементу многомерного массива. Операции с массивом и элементами массива. Основные свойства и методы. Массивы ссылок на массивы. Объявление, создание и инициализация. Доступ к элементу. Операции с элементами массива ссылок на массивы. Массив с элементами типа object. Упаковка и распаковка значений. Динамическая идентификация типов. Строка символов – объект типа string. Операции присваивания, сравнения и конкатенации строк. Преобразование строки в значение базового типа и преобразование значения базового типа в строку. Строки при вводе-выводе. Схема ввода данных с помощью средств библиотеки .NET Framework. Возможности библиотечного метода System.TryParse(). Контроль правильности вводимых данных. Объявление и инициализация строк. Доступ к элементу строки. Постоянство строк. Форматирование строки. Использование форматирования при выводе строк. Основные методы обработки строк: сравнение, поиск в строке, замена символов, вставка строк, удаление, разбиение строки на слова, формирование строки из слов, преобразование символов строки к заданному виду. Преобразование строки в массив символов и обратное преобразование. Массив строк. Параметры метода Main(). Запуск программы с передачей аргументов в метод Main().

*Тема 1.5. Методы как основа процедурного программирования*

Методы–процедуры и методы–функции. Соотношение фиксированных параметров и аргументов. Время жизни параметров. Локальные переменные и локальные константы. Параметры, передаваемые по значению и параметры, передаваемые по ссылке. Параметры с типами ссылок. Выходные параметры. Умалчиваемые значения параметров. Именованные аргументы. Модификатор params для методов с переменным числом аргументов. Перегрузка методов. Сигнатура метода при перегрузке. Рекурсивные методы. Стековые фреймы. Хвостовая рекурсия. Особенности применения метода Array.Sort

*Тема 1.6. Класс как контейнер статических членов*

Определение собственных классов. Статические члены класса. Доступность статических членов (полей и методов) из других классов. Статические поля только для чтения. Поля классов (статические поля). Статические константы. Статические методы. Статический конструктор. Правила инициализации статических полей. Статические классы.

### *Тема 1.7. Класс как тип. Объекты и их члены*

Объявление класса. Поля объектов. Объявления методов объектов. Инкапсуляция полей и методов. Средства определения доступности полей и методов из других классов. Поля только для чтения. Конструктор экземпляров класса. Конструктор умолчания. Конструктор с параметрами. Перегрузка конструкторов. Поверхностное и глубокое копирование объектов. Конструктор копирования. Свойства классов как средство доступа к закрытым полям. Определение свойств, доступных по чтению и по записи. Ссылка `this`. Деструкторы и финализаторы. Принцип инкапсуляции и методы объектов. Автореализуемые свойства. Индексаторы. Индексаторы, имитирующие наличие в объекте контейнера. Массивы объектов классов, определяемых программистом. Перегрузка операций. Декларации операций в классах и структурах. Синтаксис перегрузки унарных и бинарных операций. Перегрузка `true` и `false`.

### *Тема 1.8. Исключения*

Классические способы обработки ошибок. Предварительная проверка параметров методов. Проверка кодов возвратов методов. О механизме исключений. Системные исключения и их обработка. Свойства исключений. Исключения при арифметических операциях. Перехват исключения. Блок обработки исключения. Использование нескольких блоков обработки исключения. Передача исключений во вложенных блоках. Передача исключений между методами. Управление программой с помощью исключений. Создание собственных исключений. Понятие объекта-исключения. Пользовательские классы исключений. Ретрансляция исключения.

### *Тема 1.9. Регулярные выражения*

Регулярные выражения в командных языках и в языках программирования. Метасимволы и терминальные символы. Экранирование метасимволов. Подмножества метасимволов. Директивы нулевой ширины. Квантификаторы жадные и ленивые. Группы регулярных выражений. Замены в тексте и деление текста на части. Особенности параметров метода `Replace`.

### *Тема 1.10. Перечисления и структуры – типы значений*

Перечисления. Базовый класс перечислений. Структуры: определение типа, объявление переменных, операции над структурами. Отличие структур от классов. Упаковка и распаковка. Реализация структурами интерфейсов. Перегрузка операций в структурах.

## Раздел 2. Визуальное проектирование приложений Windows Forms

### *Тема 2.1. Визуальное проектирование приложений Windows Forms*

Формы, компоненты и элементов управления. Визуальное проектирование формы. Стандартные активные элементы графического пользовательского интерфейса: кнопки, флажки с зависимой фиксацией, флажки с независимой фиксацией, полосы прокрутки, панель индикации. Добавление элементов управления. Позиционирование элементов управления. Установка свойств формы и ее элементов. Связывание элементов формы со стандартными обработчиками событий. События от элементов формы, события клавиатуры и знакоординатных устройств (мышь). Программное изменение свойств элементов. Неотображаемые элементы (компоненты): таймер, всплывающие подсказки. Обработка событий от неотображаемых элементов. Многооконный интерфейс. Меню. Создание и отображение формы. Передача данных в подчиненную форму. Прием данных из подчиненной формы. Диалоговые окна.

### *Тема 2.2. Основы графики Windows Forms*

Организация графического вывода. Контекст отображения. Рисование на форме и в поле элемента управления. Идентификатор окна. Проблема перерисовки изображения и способы ее решения. Отслеживание состояния кнопок знако-координатного устройства. Отслеживание перемещения знако-координатного устройства. Рисование графических примитивов: линии, прямоугольники и многоугольники, эллипсы, сегменты, кривые, окрашенные фигуры. Растровые и векторные изображения. Буксировка, загрузка, рисование. Инструменты рисования: перья, кисти, шрифты. Вывод текста в графическом режиме.

### *Тема 2.3. Отношения между классами*

Включение, вложение и наследование классов. Включение объектов классов. Отношение агрегации между классами. Отношение композиции. Вложение классов. Наследование классов. Доступность членов класса при наследовании. Методы при наследовании. Копирование полей при присваивании. Конструктор умолчания. Абстрактные методы и абстрактные классы. Опечатанные классы и методы. Применение абстрактных классов. Наследование полей и методов. Порядок вызова конструкторов при создании объекта класса наследника. Служебное слово `base`. Передача аргументов конструктору базового класса. Переопределение полей. Переопределение методов. Виртуальные методы. Полиморфизм и его виды.

### *Тема 2.4. Делегаты, анонимные методы, лямбда-выражения, события и таймеры*

Синтаксис делегатов. Массивы делегатов. Многоадресные экземпляры делегатов. Делегаты и обратные вызовы. Анонимные методы. Лямбда-выражения. Объявление события. Генерация и обработка событий. Типовая структура приложения, управляемого событиями.

### *Тема 2.5. Интерфейсы*

Два вида наследования в ООП. Понятие интерфейса. Объявления интерфейсов. Реализация интерфейсов. Интерфейс как тип. Интерфейсы и наследование. Члены интерфейса: методы, свойства, события, индексаторы. Доступ к членам интерфейса. Реализация интерфейса в классе. Использование интерфейсов как альтернатива множественному наследованию

## Раздел 3. Обобщенное программирование

### *Тема 3.1. Обобщенное программирование*

Декларация обобщенных классов, интерфейсов и структур. Типизирующие параметры и список ограничений типизирующих параметров. Обобщенные методы и их сигнатуры. Виртуальные и абстрактные обобщенные методы. Вызовы обобщенных методов. Обобщенные коллекции. Обобщенные делегаты. Перегрузка операций в обобщенных классах. Использование обобщенных методов с делегатами.

### *Тема 3.2. Коллекции и итераторы*

Коллекции и их отличие от массивов. Стандартные интерфейсы. Итераторы и оператор `foreach`. Динамический массив: объявление, доступ к элементам, методы добавления и удаления элементов. Стек: объявление, доступ к элементам, методы добавления и удаления элементов. Очередь: объявление, доступ к элементам, методы добавления и удаления элементов. Хэш-таблица: объявление, доступ к элементам, методы добавления и удаления элементов. Использование коллекций при решении прикладных задач.

### *Тема 3.3. Поточный ввод-вывод*

Понятие потока данных. Стандартные потоки: стандартный поток ввода, стандартный поток вывода, стандартный поток вывода сообщений об ошибках. Символьные потоки дан-

ных, связанные с файлами. Открытие и закрытие потока. Методы чтения и записи текстовых данных. Методы чтения и записи двоичных данных. Кодировка символов в потоках: кодовые страницы, кодировка UNICODE, кодировка в текстовых потоках, кодировка текстовых строк в двоичных потоках. Буферизация потоков: буферизация двоичных потоков, буферизация текстовых потоков, принудительный сброс буферов. Потоки, ориентированные на байты. Открытие и закрытие потока. Методы чтения и записи данных. Потоки в основной памяти: создание потока, методы чтения и записи данных. Сериализация. Сериализация на основе атрибутов. Открытие потока. Определение объекта форматирования. Двоичное форматирование и XML-форматирование (двоичная сериализация и XML-сериализация)

#### *Тема 3.4. XAML и Технология WPF*

XAML (XML for Application Markup Language — XML для языка разметки приложений). Элементы и атрибуты в XAML. Свойства как атрибуты и как элементы. Зависимые и присоединенные свойства (dependency property). Расширения разметки (markup extensions). Иерархия классов WPF. Фигуры (shapes) — центральные элементы WPF. Элементы управления WPF. Компоновки элементов управления. Обработка событий в WPF. Стили, шаблоны и ресурсы. Триггеры свойств и событий. Привязка данных; команды; анимация; 3-D графика; интеграция WPF с Windows Forms.

### Раздел 4. Дополнительные возможности языка C#

#### *Тема 4.1. Язык интегрированных запросов LINQ и элементы функционального программирования*

Потоковые последовательности данных. Операции (директивы) языка LINQ. LINQ-провайдеры. Методы, используемые в LINQ-запросах. Две формы записи LINQ-запроса. Переменные запросов и разделы LINQ-запросов. Отложенное выполнение LINQ-запросов. Анонимные типы в LINQ-запросах. Операции стандартных запросов. Делегаты как параметры методов, представляющих операции запросов. Предопределенные типы делегатов в .NET. Лямбда-выражения в качестве аргументов. Функции высших порядков.

#### *Тема 4.2. Асинхронные методы и параллельные программы*

Процессы операционной системы и потоки исполнения. Многопоточность. Пул потоков исполнения. Класс Thread и его члены. Создание потоков. Синхронизация потоков. Передача данных в поток и из потока. Обмены между потоками. Патерны асинхронного программирования: патерн опроса (polling), патерн ожидания, патерн ответного вызова. Механизм async/await. Возможности делегатов в параллельном программировании. Таймеры.

#### *Тема 4.3. Небезопасный код и препроцессорные директивы C#*

Небезопасный код. Указатели и адресная арифметика. Препроцессорная обработка. Препроцессорные директивы.

#### *Тема 4.4. Мобильные приложения и Web-приложения*

Windows Store Apps. Браузерное приложение WPF (XAML Browser Applications).

##### 5.2. Лабораторные работы

Лабораторная работа №1. Основные элементы программ (данные, выражения, операторы)

Лабораторная работа № 2. Массивы и строки

Лабораторная работа № 3. Методы как основа процедурного программирования

Лабораторная работа № 4. Класс как контейнер статических членов

Лабораторная работа № 5. Класс как тип. Объекты и их члены

Лабораторная работа №6. Исключения

- Лабораторная работа №7. Регулярные выражения  
 Лабораторная работа №8. Перечисления и структуры – типы значений  
 Лабораторная работа №9. Визуальное проектирование приложений Windows Forms  
 Лабораторная работа №10. Основы графики Windows Forms  
 Лабораторная работа №11. Отношения между классами  
 Лабораторная работа №12. Делегаты, анонимные методы, лямбда-выражения, события и таймеры  
 Лабораторная работа №13. Интерфейсы  
 Лабораторная работа №14. Обобщенное программирование  
 Лабораторная работа №15. Коллекции и итераторы  
 Лабораторная работа №16. Поточковый ввод-вывод  
 Лабораторная работа №17. XAML и Технология WPF  
 Лабораторная работа №18. Язык интегрированных запросов LINQ и элементы функционального программирования  
 Лабораторная работа №19. Асинхронные методы и параллельные программы  
 Лабораторная работа №20. Небезопасный код и препроцессорные директивы C#  
 Лабораторная работа №21. Мобильные приложения и Web-приложения

5.3. Вопросы для самостоятельной работы студента в соответствии с содержанием разделов дисциплины

1. Типы приложений.
2. Структура исходного кода консольного приложения.
3. Битовые операции.
4. Массив строк.
5. Хвостовая рекурсия.
6. Правила инициализации статических полей.
7. Синтаксис перегрузки унарных и бинарных операций.
8. Ретрансляция исключения.
9. Особенности параметров метода Replace.
10. Перегрузка операций в структурах.
11. Передача данных в подчиненную форму.
12. Прием данных из подчиненной формы.
13. Вывод текста в графическом режиме.
14. Полиморфизм и его виды.
15. Типовая структура приложения, управляемого событиями.
16. Использование интерфейсов как альтернатива множественному наследованию.
17. Использование обобщенных методов с делегатами.
18. Использование коллекций при решении прикладных задач
19. Двоичное форматирование и XML- форматирование (двоичная сериализация и XML-сериализация).
20. Привязка данных; команды; анимация; 3-D графика; интеграция WPF с Windows Forms.
21. Лямбда-выражения в качестве аргументов. Функции высших порядков.
22. Возможности делегатов в параллельном программировании. Таймеры.
23. Препроцессорные директивы.

## 6. Образовательные технологии

В соответствии со структурой образовательного процесса по дисциплине применяются следующие технологии:

- диагностики;

- целеполагания;
- управления процессом освоения учебной информации;
- применения знаний на практике, поиска новой учебной информации;
- организации совместной и самостоятельной деятельности обучающихся (учебно-познавательной, научно-исследовательской, частично-поисковой, репродуктивной, творческой и пр.);

В соответствии с требованиями ФГОС ВО для реализации компетентностного подхода при обучении дисциплине предусмотрено широкое использование в учебном процессе активных и интерактивных методов проведения занятий:

Интерактивные технологии:

№ темы	Вид занятия	Используемые интерактивные технологии	Всего часов
1-23	Лабораторная работа	Метод проектов	88

При обучении дисциплине применяются следующие формы занятий:

- лекции, направленные на получение новых и углубление научно-теоретических знаний, в том числе вводная лекция, информационная лекция, обзорная лекция, лекция-консультация, проблемная лекция, лекции-дискуссии, лекции-беседы и др.;
- лабораторные занятия, проводимые под руководством преподавателя в учебной лаборатории с использованием компьютеров и учебного оборудования, направленные на закрепление и получение новых умений и навыков, применение знаний и умений, полученных на теоретических занятиях, при решении практических задач и др.

Все занятия обеспечены мультимедийными средствами (SMART доски, проекторы, экраны) для повышения качества восприятия изучаемого материала. В образовательном процессе широко используются информационно-коммуникационные технологии.

Самостоятельная работа студентов – это планируемая работа студентов, выполняемая по заданию при методическом руководстве преподавателя, но без его непосредственного участия. Формы самостоятельной работы студентов определяются содержанием учебной дисциплины, степенью подготовленности студентов. Они могут иметь учебный или учебно-исследовательский характер: подготовка к лабораторным работам.

Формами контроля самостоятельной работы выступают: проверка письменных отчетов по результатам выполненных лабораторных работ. Результаты самостоятельной работы учитываются при оценке знаний на зачете и экзамене.

## 7. Формы аттестации и оценочные материалы

Текущий контроль предназначен для проверки хода и качества усвоения учебного материала, стимулирования учебной работы обучающихся и совершенствования методики проведения занятий. Он проводится в ходе всех видов занятий в форме, избранной преподавателем.

Промежуточная аттестация имеет целью определить степень достижения учебных целей по учебной дисциплине и проводится в форме экзамена, зачета. Принимается экзамен и зачет преподавателем, читающим лекции по данной учебной дисциплине в соответствии с перечнем основных вопросов, выносимых для контроля знаний обучающихся:

### 7.1. Вопросы к зачету

*Вопросы (5 семестр):*

1. Введение. Императивная и декларативная парадигмы программирования. Эволюция языков программирования
2. Принципы программирования на основе платформы .Net Framework

3. Основные элементы программ (данные, выражения, операторы)
4. Массивы и строки
5. Методы как основа процедурного программирования
6. Класс как контейнер статических членов
7. Класс как тип. Объекты и их члены
8. Исключения
9. Регулярные выражения
10. Перечисления и структуры – типы значений

*Вопросы (7 семестр):*

1. Обобщенное программирование
2. Коллекции и итераторы
3. Поточковый ввод-вывод
4. XAML и технология WPF

Зачет проводится по окончании занятий по дисциплине до начала экзаменационной сессии.

Билет для проведения промежуточной аттестации в форме зачета включают вопросы для проверки сформированности знаний, умений и навыков.

- Оценка «зачтено» проставляется студенту, выполнившему и защитившему в полном объеме лабораторные работы в течение семестра, чей уровень знаний, умений и навыков соответствует уровню оценок «отлично», «хорошо» или «удовлетворительно».
- Оценка «не зачтено» проставляется студенту, не выполнившему и (или) не защитившему в полном объеме практические задания и лабораторные работы в течение семестра, либо чей уровень знаний, умений и навыков соответствует уровню оценки «неудовлетворительно».

Общими критериями, определяющими оценку знаний, умений и навыков являются:

- для оценки «отлично» - наличие глубоких и исчерпывающих знаний в объеме пройденного программного материала правильные и уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, знание дополнительно рекомендованной литературы;
- для оценки «хорошо» - наличие твердых и достаточно полных знаний программного материала, незначительные ошибки при освещении заданных вопросов, правильны действия по применению знаний на практике, четкое изложение материала;
- для оценки «удовлетворительно» - наличие твердых знаний пройденного материала, изложение ответов с ошибками, уверенно исправляемыми после дополнительных вопросов, необходимость наводящих вопросов, правильные действия по применению знаний на практике;
- для оценки «неудовлетворительно» - наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

*7.2. Вопросы и задачи к экзамену*

*Вопросы (6 семестр)*

1. Введение. Императивная и декларативная парадигмы программирования. Эволюция языков программирования
2. Принципы программирования на основе платформы .Net Framework
3. Основные элементы программ (данные, выражения, операторы)
4. Массивы и строки

5. Методы как основа процедурного программирования
6. Класс как контейнер статических членов
7. Класс как тип. Объекты и их члены
8. Исключения
9. Регулярные выражения
10. Перечисления и структуры – типы значений
11. Визуальное проектирование приложений Windows Forms
12. Основы графики Windows Forms
13. Отношения между классами
14. Делегаты, анонимные методы, лямбда-выражения, события и таймеры
15. Интерфейсы

*Темы задач(6 семестр):*

*Определение классов. Наследование. Виртуальные функции*

1. Создать класс «Треугольник», заданный координатами вершин. Определить производные классы «Равносторонний» и «Прямоугольный» треугольники. Предусмотреть виртуальную функцию для вычисления площади описанной около треугольника окружности, сравнить площади окружностей, описанных около равностороннего и прямоугольного треугольников, координаты которых введены с клавиатуры.

2. Создать класс «Окружность», заданный координатами центра и точкой на окружности. Определить производный класс «Описанная окружность». Предусмотреть виртуальную функцию для вычисления периметра равностороннего треугольника, вписанного в окружность. Сравнить периметры 3 равносторонних треугольников, вписанных в окружности, данные о которых вводятся с клавиатуры.

3. Создать класс «Окружность», заданный координатами центра и точкой на окружности. Определить производный класс «Описанная окружность». Предусмотреть виртуальную функцию для вычисления площади равностороннего треугольника, вписанного в окружность. Сравнить площади 2 равносторонних треугольников, вписанных в окружности, данные о которых вводятся с клавиатуры.

4. Создать класс «Шар», заданный координатами центра и точкой на сфере. Определить производный класс «Окружность». Предусмотреть виртуальную функцию для вычисления площади круга, сравнить площади 2 кругов, данные о которых введены с клавиатуры.

5. Создать класс «Шар», заданный радиусом. Определить производный класс «Окружность». Предусмотреть виртуальную функцию для вычисления площади круга и длины окружности, сравнить площади и длины 2 окружностей, радиусы которых введены с клавиатуры.

*Вопросы (8 семестр):*

1. Введение. Императивная и декларативная парадигмы программирования. Эволюция языков программирования
2. Принципы программирования на основе платформы .Net Framework
3. Основные элементы программ (данные, выражения, операторы)
4. Массивы и строки
5. Методы как основа процедурного программирования
6. Класс как контейнер статических членов
7. Класс как тип. Объекты и их члены
8. Исключения
9. Регулярные выражения
10. Перечисления и структуры – типы значений
11. Визуальное проектирование приложений Windows Forms
12. Основы графики Windows Forms
13. Отношения между классами
14. Делегаты, анонимные методы, лямбда-выражения, события и таймеры
15. Интерфейсы



16. Обобщенное программирование
17. Коллекции и итераторы
18. Поточковый ввод-вывод
19. XAML и технология WPF
20. Язык интегрированных запросов LINQ и элементы функционального программирования
21. Асинхронные методы и параллельные программы
22. Небезопасный код и препроцессорные директивы C#
23. Мобильные приложения и Web-приложения

*Темы задач (8 семестр):*

*Поточковый ввод-вывод. Текстовые и бинарные потоки. Перегрузка ввода-вывода*

1. В одном файле хранятся сведения о книгах: код, автор, название, во 2 файле сведения об издании: код, издательство, год издания. Вывести на экран информацию о книгах автора, введенного с клавиатуры, изданных в 2013 году, введенного с клавиатуры (ФИО, должность, зарплата за год), выбрать сотрудников, получивших зарплату за год меньше средней по отделу.

2. В одном файле хранятся сведения о сотрудниках: ФИО, табельный номер, должность, отдел, во 2 файле сведения о работе за год: табельный номер, заработная плата по месяцам. Вывести на экран информацию о сотрудниках отдела, введенного с клавиатуры (ФИО, должность, зарплата за год), выбрать сотрудников, получивших зарплату за год меньше средней по отделу.

3. В одном файле хранятся сведения о товарах в магазине: код товара, название, цена, во 2 файле сведения о производстве: код товара, производитель, дата производства, срок годности. Вывести на экран информацию о товарах произведенных после 20.06.2012 (название, цена, производитель, дата производства), которые стоят менее 300 рублей, у которых истек срок годности.

4. В одном файле хранятся сведения о сотрудниках: ФИО, табельный номер, должность, отдел, во 2 файле сведения о работе за год: табельный номер, заработная плата по месяцам. Вывести на экран информацию о сотрудниках отдела, введенного с клавиатуры (ФИО, должность, зарплата за год), отсортированную по фамилиям в алфавитном порядке.

5. В одном файле хранятся сведения об учениках: ФИО, класс, во 2 файле сведения о результатах сдачи экзаменов: ФИО, название предмета, ФИО преподавателя, оценка. Вывести на экран фамилии, класс и оценки учеников, сдавших математику, но не сдавших русский язык.

Билет для проведения промежуточной аттестации в форме экзамена включает вопросы и задачи для проверки сформированности знаний, умений и навыков.

Общими критериями, определяющими оценку знаний, умений и навыков являются:

- для оценки «отлично» - наличие глубоких и исчерпывающих знаний в объеме пройденного программного материала правильные и уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, знание дополнительно рекомендованной литературы;

- для оценки «хорошо» - наличие твердых и достаточно полных знаний программного материала, незначительные ошибки при освещении заданных вопросов, правильны действия по применению знаний на практике, четкое изложение материала;

- для оценки «удовлетворительно» - наличие твердых знаний пройденного материала, изложение ответов с ошибками, уверенно исправляемыми после дополнительных вопросов, необходимость наводящих вопросов, правильные действия по применению знаний на практике;

- для оценки «неудовлетворительно» - наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

### 7.3. Выполнение и примерная тематика курсового проекта

Курсовой проект не предусмотрен.

## 7.4. Выполнение и примерные задания расчетно-графической работы

Расчетно-графическая работа не предусмотрена.

## 7.5. Выполнение и примерная тематика контрольной работы

Контрольная работа не предусмотрена.

**8. Учебно-методическое и информационное обеспечение дисциплины**

Электронный каталог и электронные информационные ресурсы, предоставляемые научной библиотекой ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова» доступны по ссылке <http://library.chuvsu.ru/>

8.1. Рекомендуемая основная литература (ежегодное обновление перечня и условия доступа представлены в Приложениях к рабочей программе)

№ п/п	Наименование
1.	Котов О.М. Язык C#. Краткое описание и введение в технологии программирования [Электронный ресурс] : учебное пособие / О.М. Котов. — Электрон. текстовые данные. — Екатеринбург: Уральский федеральный университет, 2014. — 208 с. - Режим доступа: <a href="http://www.iprbookshop.ru/68524.html">http://www.iprbookshop.ru/68524.html</a>
2.	Никишев, В. К. Объектно-ориентированное программирование : учебное пособие / В. К. Никишев ; [отв. ред. В. П. Желтов] ; Чуваш. гос. ун-т им. И. Н. Ульянова. - Чебоксары : Изд-во Чуваш. ун-та, 2016. - 262с.
3.	Шацков В.В. Программирование приложений баз данных с использованием СУБД MS SQL Server [Электронный ресурс] : учебное пособие / В.В. Шацков. — Электрон. текстовые данные. — СПб. : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2015. — 80 с. – Режим доступа: <a href="http://www.iprbookshop.ru/63638.html">http://www.iprbookshop.ru/63638.html</a>

8.2. Рекомендуемая дополнительная литература (ежегодное обновление и условия доступа перечня представлены в Приложениях к рабочей программе)

№ п/п	Наименование
1.	Никишев, В. К. Информатика и программирование : учебное пособие / В. К. Никишев ; [отв. ред. В. П. Желтов] ; Чуваш. гос. ун-т им. И. Н. Ульянова. - Чебоксары : Изд-во Чуваш. ун-та, 2015. - 220с.
2.	Разработка Windows-приложений в среде программирования Visual Studio.Net [Электронный ресурс] : учебно-методическое пособие по дисциплине Информатика и программирование / . — Электрон. текстовые данные. — М. : Московский технический университет связи и информатики, 2016. — 20 с. - Режим доступа: <a href="http://www.iprbookshop.ru/61536.html">http://www.iprbookshop.ru/61536.html</a>
3.	Обломов, И. А. Объектно-ориентированное программирование : лабораторный практикум / И. А. Обломов ; [отв. ред. А. Л. Симаков] ; Чуваш. гос. ун-т им. И. Н. Ульянова. - Чебоксары : Изд-во Чуваш. ун-та, 2014. - 111с.

8.3. Рекомендуемые методические разработки по дисциплине

№ п/п	Наименование	Условия доступа
1.	ООП: лабораторный практикум	URL: <a href="http://moodle.chuvsu.ru/course/view.php?id=1208">http://moodle.chuvsu.ru/course/view.php?id=1208</a>

8.4. Программное обеспечение, профессиональные базы данных, информационно-справочные системы.

Доступное программное лицензионное и свободно распространяемое программное обеспечение, предоставляемое студенту университетом возможно для загрузки и использования по URL: <http://ui.chuvsu.ru/> \*.

№ п/п	Наименование Рекомендуемого ПО	Условия доступа/скачивания
		свободное лицензионное соглашение:
1.	Microsoft Visual Studio	<a href="https://www.microsoft.com/ru-ru/SoftMicrosoft/vs2017">https://www.microsoft.com/ru-ru/SoftMicrosoft/vs2017</a>
2.	DevC++	<a href="https://sourceforge.net/projects/orwelldevcpp/">https://sourceforge.net/projects/orwelldevcpp/</a>
3.	Linux/ Ubuntu	<a href="http://ubuntu.ru/">http://ubuntu.ru/</a>
4.	LibreOffice	<a href="https://ru.libreoffice.org/">https://ru.libreoffice.org/</a>
1.	Microsoft Windows	из внутренней сети университета (договор)*
2.	Microsoft Office	
1.	Гарант	из внутренней сети университета (договор)*
2.	Консультант +	

#### 8.5. Рекомендуемые интернет-ресурсы и открытые он-лайн курсы

№ п/п	Наименование интернет ресурса	Режим доступа
1.	Открытое образование. Программирование на C#	URL: <a href="https://openedu.ru/course/urfu/CSHARP/">https://openedu.ru/course/urfu/CSHARP/</a>
2.	Национальный открытый университет. Создание Windows-приложений на основе Visual C#	URL: <a href="http://www.intuit.ru/studies/courses/106/106/info">http://www.intuit.ru/studies/courses/106/106/info</a>
3.	Национальный открытый университет. Программирование на языке высокого уровня C#	URL: <a href="http://www.intuit.ru/studies/courses/629/485/info">http://www.intuit.ru/studies/courses/629/485/info</a>

### 9. Материально-техническое обеспечение дисциплины

Учебные аудитории для лекционных занятий по дисциплине оснащены автоматизированным рабочим местом (АРМ) преподавателя, обеспечивающим тематические иллюстрации и демонстрации, соответствующие программе дисциплины в составе:

- ПЭВМ с доступом в интернет (операционная система, офисные программы, антивирусные программы);
- мультимедийный проектор с дистанционным управлением;
- настенный экран;

Учебные аудитории для лабораторных и самостоятельных занятий по дисциплине оснащены АРМ преподавателя и пользовательскими АРМ по числу обучающихся, объединенных локальной сетью («компьютерный» класс), с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова».

### 10. Средства адаптации преподавания дисциплины к потребностям лиц с ограниченными возможностями

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.
- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.
- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

## **11. Методические рекомендации по освоению дисциплины**

В ходе лекционных занятий студенту рекомендуется вести конспектирование учебного материала. Следует обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации, положительный опыт в ораторском искусстве. При составлении конспекта желательно оставлять в рабочих конспектах поля, на которых в дальнейшем можно делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. В ходе лекционных занятий рекомендуется задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

В ходе подготовки к лабораторным работам рекомендуется изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях. Основой для выполнения лабораторной работы являются разработанные кафедрой методические указания. Рекомендуется дорабатывать свой конспект лекций, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной рабочей программой дисциплины.

Форма организации студентов на лабораторных работах индивидуальная. При индивидуальной форме организации занятий каждый студент выполняет индивидуальное задание.

Если в результате выполнения лабораторной работы запланирована подготовка письменного отчета, то отчет о выполненной работе необходимо оформлять в соответствии с требованиями методических указаний. Качество выполнения лабораторных работ является важной составляющей оценки текущей успеваемости обучающегося.