

**Федеральное государственное автономное образовательное учреждение  
высшего образования  
"Национальный исследовательский университет  
"Высшая школа экономики"**

Факультет экономики, менеджмента и бизнес информатики  
Кафедра информационных технологий в бизнесе

**Рабочая программа дисциплины Программирование**

для образовательной программы Программная инженерия  
направления подготовки 09.03.04 Программная инженерия  
уровень бакалавр

Разработчик программы  
Викентьева О.Л. , ovikenteva@hse.ru

Одобрена на заседании кафедры информационных технологий в бизнесе  
«30» августа 2017 г.  
И.о. зав. кафедрой  
О.Л. Викентьева \_\_\_\_\_

Утверждена Академическим советом образовательной программы «Бизнес-информатика»  
направления подготовки 38.03.05 Бизнес-информатика, образовательной программы  
«Программная инженерия» направления подготовки 09.03.04 Программная инженерия,  
образовательной программы «Информационная аналитика в управлении предприятием»  
направления подготовки 38.04.05 Бизнес-информатика.  
«31» августа 2017 г., № протокола 8.2.2.1 – 35–09/04

Академический руководитель образовательной программы «Программная инженерия»  
А.О. Сухов \_\_\_\_\_

Пермь, 2017

*Настоящая программа не может быть использована другими подразделениями  
университета и другими вузами без разрешения подразделения-разработчика программы.*



## 1. Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов направления подготовки 09.03.04 Программная инженерия, изучающих дисциплину «Программирование».

Программа разработана в соответствии с:

- Образовательным стандартом НИУ ВШЭ по направлению подготовки 09.03.04 Программная инженерия, утвержденным Ученым советом Национального исследовательского университета «Высшая школа экономики», протокол от 30.01.2015 № 1 в редакции 2017 г. (утверждена профессиональной коллегией УМС НИУ ВШЭ, протокол от 14.06.2017 № 2);
- Учебным планом университета по направлению подготовки 09.03.04 Программная инженерия, утвержденным в 2016 г.

## 2. Цели освоения дисциплины

Целями освоения дисциплины Программирование являются:

- формирование у студентов алгоритмического мышления;
- приобретение теоретических знаний и практических навыков в области основных методологий разработки программ с помощью языков программирования высокого уровня (процедурно-ориентированной и объектно-ориентированной);
- изучение современных инструментальных сред, предназначенных для разработки программ с помощью языков программирования высокого уровня;

## 3. Компетенции обучающегося, формируемые в результате освоения дисциплины

Бакалавр по направлению подготовки 09.03.04 Программная инженерия в соответствии с образовательным стандартом должен знать основные технологии программирования, владеть методами и инструментальными средствами разработки программ. Основы этих компетенция закладываются при изучении данной дисциплины.

В результате освоения дисциплины студент должен:

- *Знать:*
  - основные понятия информатики и программирования;
  - основные управляющие структуры и способы описания алгоритмов;
  - основные методы разработки алгоритмов, особенности их реализации;
  - понятие типа данных, форматы представления данных при решении задач с помощью компьютера, а также средства конструирования новых типов на основе стандартных типов, используемых в языках программирования;
  - основы организации файлов, выполнения операций над файлами, возможности их использования при решении задач;
  - методы организации, хранения и эффективной обработки абстрактных структур данных;
  - основные технологии программирования: процедурно-ориентированное и объектно-ориентированное программирование с использованием языка высокого уровня;



- *Уметь:*
  - решать задачи, используя различные методы разработки алгоритмов и выбирая наиболее подходящие алгоритмы и средства их реализации в зависимости от постановки задачи;
  - разрабатывать программные продукты: разрабатывать программы средней сложности на языках программирования высокого уровня;
  - использовать основные концепции процедурного и объектно-ориентированного программирования; отлаживать и тестировать программы.
- *Владеть (приобрести опыт):*
  - навыками разработки и анализа алгоритмов решения типовых задач (сортировки и поиска данных, поиска решений и пр.), исследования их свойств;
  - методами и инструментальными средствами разработки программ: разработки программ средней сложности, их тестирования и отладки;
  - навыками самостоятельного решения задач с помощью компьютеров, изучения новых средств разработки программ.

Уровни формирования компетенций:

**РБ** — ресурсная база, в основном теоретические и предметные основы (знания, умения);

**СД** – способы деятельности, составляющие практическое ядро данной компетенции;

**МЦ** – мотивационно-ценностная составляющая, отражает степень осознания ценности компетенции человеком и готовность ее использовать

В результате освоения дисциплины студент осваивает компетенции:

Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
Способен учиться, приобретать новые знания, умения, в том числе в области, отличной от профессиональной	УК-1	РБ	Знает определения основных понятий, может выбрать способ решения задачи, умеет анализировать условие задачи, разрабатывать ее алгоритм и тесты (черный ящик, МГТ).	Разбор примеров во время лекционных занятий, выполнение практических заданий, связанных с анализом предметной области, проектированием и разработкой программ, их тестирование с использованием различных методов и отладки	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе
Способен критически оценивать и переосмысливать накопленный	УК-9	РБ	Способен читать и анализировать программный код,	Разбор примеров во время лекционных занятий, выполнение	защита отчетов по практическим работам, сдача отчетов по самостоятельной



Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
опыт (собственный и чужой), рефлексировать профессиональную и социальную деятельность			написанный другими программистами, находить в нем ошибки	практических заданий, связанных с анализом программного кода, написанного другими программистами, поиск ошибок, рефакторинг кода.	работе
Способен применять основные концепции, принципы, теории и факты, связанные с информатикой при решении научно-исследовательских задач	ПК-1	РБ	Способен разрабатывать программы на основе основных концепций, принципов, теорий и фактов, связанные с информатикой	Разбор примеров во время лекционных занятий, выполнение практических заданий, связанных с анализом предметной области, разработкой алгоритмов и программ.	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе
Способен к формализации в своей предметной области с учетом ограничений используемых методов исследования	ПК-2	РБ	Применяет различные методы для анализа и разработки алгоритмов и программ	Выполнение заданий, связанных с анализом предметной области и разработкой программ.	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе
Способен использовать методы и инструментальные средства исследования объектов профессиональной деятельности	ПК-3	СД	Применяет различные методы и инструментальные средства для анализа и разработки алгоритмов и программ	Выполнение заданий, связанных с анализом предметной области и разработкой программ с помощью IDE.	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе
Способен обосновать принимаемые проектные решения, осуществлять	ПК-4	РБ	Обосновывает используемые методы для анализа и разработки алгоритмов и	Выполнение заданий, связанных с анализом предметной области и	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе



Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
постановку и выполнение экспериментов по проверке их корректности и эффективности			программ	разработкой программ	
Способен готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты исследований в виде статей и докладов на научно-технических конференциях	ПК-5	РБ	Владеет навыками грамотного оформления и документирования текстов программ, результатов их тестирования	Оформление результатов выполнения заданий с использованием требований к оформлению отчетов, современных технологий подготовки документов.	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе.
Способен формализовать предметную область программного проекта и разработать спецификации для компонентов программного продукта	ПК-6	СД	Способен применять различные методы и инструментальные средства для анализа и разработки алгоритмов и программ	Выполнение заданий, связанных с анализом предметной области и разработкой программ	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе.
Способен проектировать, конструировать и тестировать программные продукты	ПК-10	СД	Знает возможности системы программирования и может разрабатывать программы средней сложности на языке программирования	Выполнение практических заданий с использованием языка программирования высокого уровня, их тестирование с использованием различных	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе.



Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
			ия высокого уровня, владеет средствами тестирования и отладки программ с использованием возможностей системы программирования	методов и отладки	
Способен использовать различные технологии разработки программного обеспечения	ПК-16	РБ	Владеет навыками разработки программ с использованием процедурно-ориентированной и объектно-ориентированной технологий программирования	Выполнение заданий, связанных с анализом предметной области разработкой программ	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе.
Способен применять основные методы и инструменты разработки программного обеспечения	ПК-17	СД	Владеет методами и инструментальными средствами разработки программ, в частности основными возможностями современных систем программирования, языков высокого уровня	Выполнение заданий, связанных с анализом предметной области, разработкой и тестированием программ.	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе.
Способен понимать стандарты и модели жизненного цикла	ПК-19	РБ	Имеет представление о жизненном цикле программных систем и его моделях.	Выполнение практических заданий, связанных с анализом предметной области разработкой программ, их тестированием с использованием различных методов и отладки,	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе.



Компетенция	Код по ОС ВШЭ	Уровень формирования компетенции	Дескрипторы – основные признаки освоения (показатели достижения результата)	Формы и методы обучения, способствующие формированию и развитию компетенции	Форма контроля уровня сформированности компетенции
				выполнение заданий в рамках командного проекта по ролям, деловая игра.	
Способен к осознанному целеполаганию, профессиональному и личностному развитию	ПК-30	МЦ	Демонстрирует навыки профессиональной коммуникации, самостоятельность при выполнении работ добросовестность, упорство, настойчивость в достижении результатов	Выполнение практических заданий, связанных с анализом предметной области и разработкой программ, их тестирование с использованием различных методов и отладки, выполнение заданий в рамках командного проекта по ролям, деловая игра.	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе.
Способен понимать и анализировать мировоззренческие, социально и личностно значимые проблемы и процессы, происходящие в обществе	ПК-33	МЦ	Демонстрирует навыки работы в рамках выбранной роли при выполнении групповой работы, помощь одноклассникам при выполнении code review, тестировании и т.п.	Выполнение заданий в рамках командного проекта по ролям, деловая игра.	защита отчетов по практическим работам, сдача отчетов по самостоятельной работе.

#### 4. Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к базовой части профессионального цикла дисциплин.

Изучение данной дисциплины базируется на следующих дисциплинах:

- Базовый школьный курс информатики
- Алгебра.
- Дискретная математика.





Основные положения дисциплины будут использованы в дальнейшем при изучении следующих дисциплин учебного плана:

- Введение в программную инженерию.
- Архитектура вычислительных систем.
- Алгоритмы и структуры данных.
- Конструирование программного обеспечения.
- Базы данных.
- Обеспечение качества и тестирование.
- Разработка и анализ требований.
- Проектирование архитектуры программных систем.
- Управление программными проектами.
- Компьютерный практикум по основам алгоритмизации и методам программирования.
- Научно-исследовательский семинар.

## 5. Тематический план учебной дисциплины

№	Название раздела	Всего часов	Аудиторные часы			Самостоятельная работа
			Лекции	Семинары	Практические занятия	
<b>1</b>	<b>Раздел 1. Основы программирования на языке программирования высокого уровня</b>	<b>76</b>	<b>14</b>		<b>18</b>	<b>44</b>
	Тема 1. Жизненный цикл программного обеспечения.	10	2		2	6
	Тема 2. Общая характеристика системы MSDN.	10	2		2	6
	Тема 3. Состав языка программирования высокого уровня.	18	2		4	12
	Тема 4. Основные операторы	18	4		4	10
	Тема 5. Процедурно-ориентированная технология разработки программ	20	4		6	10
<b>3</b>	<b>Раздел 2. Структурированные типы данных</b>	<b>70</b>	<b>10</b>		<b>18</b>	<b>42</b>
	Тема 6. Массивы, строки	16	2		4	10
	Тема 7. Динамические структуры данных (списки, бинарные деревья)	16	4		4	8
	Тема 8. Файлы	16	2		2	12
	Тема 9. Визуальное проектирование Windows приложений	22	2		8	12
<b>4</b>	<b>Раздел 3. Основные свойства ООП</b>	<b>70</b>	<b>8</b>		<b>16</b>	<b>46</b>
	Тема 10. Классы, объекты, методы. Перегрузка методов.	22	2		6	14
	Тема 11. Основные свойства ООП: инкапсуляция.	20	2		4	14
	Тема 12. Основные свойства ООП: наследование, полиморфизм.	28	4		6	18
<b>5</b>	<b>Раздел 4. Абстрактные типы данных. Обобщенное программирование.</b>	<b>44</b>	<b>8</b>		<b>14</b>	<b>22</b>
	Тема 13. Абстрактные типы данных.	28	6		10	12





	Коллекции.					
	Тема 14 Обобщенное программирование.	16	2		4	10
<b>6</b>	<b>Раздел 5. Объектно-событийное и объектно-ориентированное программирование.</b>	<b>42</b>	<b>10</b>		<b>14</b>	<b>18</b>
	Тема 15. Обработка исключительных ситуаций.	16	4		4	8
	Тема 16. OO программа как программа, управляемая событиями.	26	6		10	10
<b>7</b>	<b>Раздел 6. LINQ to objects</b>	<b>36</b>	<b>4</b>		<b>8</b>	<b>20</b>
	Тема 17. Дополнительные возможности	18	4		4	10
	Тема 18. Запросы LINQ to objects.	18	4		4	10
<b>8</b>	<b>Раздел 7. Многопоточное программирование</b>	<b>42</b>	<b>12</b>		<b>12</b>	<b>18</b>
	Тема 19. Создание потоков.	20	6		6	8
	Тема 20. Синхронизация потоков.	22	6		6	10
	<b>Итого</b>	<b>380</b>	<b>70</b>		<b>100</b>	<b>210</b>

## 6. Формы контроля знаний студентов

Тип контроля	Форма контроля	1 год				Параметры
		1	2	3	4	
Текущий	Домашнее задание		9		9	Разработка программы на языке программирования высокого уровня
	Самостоятельная работа	2,4,6	2,4,6	1,3,5,7,9	1,3,5,7,9	Оформление отчетов по лабораторным работам
	Лабораторная работа	2,4,6	2,4,6	1,3,5,7	1,3,5,7	Разработка программы на языке программирования высокого уровня
Итоговый	Экзамен				*	Письменный тест (80 минут)

## 7. Критерии оценки знаний, навыков

В рамках текущего контроля студент должен продемонстрировать, что он:

- чётко формулирует задачи, анализирует условия и обоснованно выбирает методы решения, уверенно интерпретирует полученные результаты;
- владеет методами и средствами разработки программ, в частности основными возможностями современных систем программирования, языков высокого уровня:
  - знает возможности системы программирования и может разрабатывать программы средней сложности на языке программирования высокого уровня;
  - владеет средствами тестирования и отладки программ с использованием возможностей системы программирования;
- способен самостоятельно определять формирующиеся дефициты знаний, умений и навыков в ходе обучения, может сформулировать проблемы, связанные с недостатком знаний и навыков, и выбрать подходы к их решению;
- владеет навыками поиска и использования информации, необходимой для выполнения заданий (поиск описаний алгоритмов, методов их оценки и пр.), из различных источников;
- умеет самостоятельно работать со справочной информацией, руководствами, написанными на английском языке, и владеет знаниями, достаточными для



- самостоятельного изучения и понимания описаний алгоритмов и программ, формальных моделей, полученными из источников на английском языке;
- способен обосновывать предлагаемые решения (не только разрабатывать алгоритмы и программы, реализующие их, но и уметь доказывать правильность программ, анализировать и оценивать эффективность решений);
- владеет навыками грамотного оформления и документирования текстов программ, результатов их тестирования;
- умеет грамотно оформлять отчеты о выполнении домашних заданий, включающие постановку задач, описание решений и оценки результатов.

Текущий контроль предусматривает выполнение лабораторных работ, оформление отчетов по выполненным лабораторным работам, а также защиту выполненной лабораторной работы. Во втором и четвертом модулях предусмотрено выполнение домашнего задания. При оценке домашнего задания учитываются оценки за лабораторные работы 1-7, При оценке домашнего задания 2 задания учитываются оценки за лабораторные работы 8-15.

Оценки по всем формам текущего контроля выставляются по 10-ти балльной шкале.

Формы и сроки проведения определяются учебным планом и графиком учебного процесса.

Итоговый контроль в четвертом модуле осуществляется в форме письменного теста и письменного решения задачи (по желанию преподавателя при решении задачи студент может использовать систему программирования MS Visual Studio).

Теоретические вопросы включают вопросы по темам всего курса, которые изучались в 3-4 модулях (по темам 1-2 модулей проверяются остаточные знания: знание основных понятий и свойств, методов и алгоритмов, умение использовать математический аппарат для решения задач).

Выдача заданий на лабораторные работы и домашнее задания осуществляется с помощью системы LMS, отчеты по лабораторным работам сдаются также при помощи системы LMS.

## **8. Содержание дисциплины**

### **Раздел I. Основы программирования на языке программирования высокого уровня.**

#### **Тема 1. Жизненный цикл программного обеспечения [10 часов]**

Постановка задачи и спецификация программы, способы записи алгоритма. Программа на языке высокого уровня. Понятия тестирования и отладки. Критерии качества программы. Диалоговые программы, дружелюбность интерфейса. Стиль программирования. Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма. Типовые алгоритмы вычислительных процессов.

Лекции: 2 часа.

Практические занятия: 2 часа.

Самостоятельная работа: 6 часов.

#### **Тема 2. Общая характеристика системы MSDN [10 часов]**

Понятия системы программирования и среды программирования. Основные понятия. Выполнение программы в .NET. Структура платформы .NET. Интегрированная среда разработки MS Visual Studio.NET. Обобщенная система типов.

Лекции: 2 часа.

Практические занятия: 2 часа.

Самостоятельная работа: 6 часов.

#### **Тема 3. Состав языка программирования высокого уровня [18 часов]**

Структура программы на языке высокого уровня. Алфавит языка. Лексемы.



Понятие типа данных. Классификация типов. Значимые типы и ссылочные типы данных. Классификация по способу хранения. Встроенные типы данных.

Переменные. Выражения. Основные операции. Использование стандартных функций. Приведение типов.

Лекции: 2 часа.

Практические занятия: 4 часа.

Самостоятельная работа: 12 часов.

#### **Тема 4. Основные операторы [18 часов]**

Операторы и основные управляющие структуры: итерация, ветвление, повторение.

Решение задач с использованием основных управляющих структур. Вложенные циклы. Рекуррентные вычисления.

Лекции: 4 часа.

Практические занятия: 4 часа.

Самостоятельная работа: 10 часов.

#### **Тема 5. Процедурно-ориентированная технология разработки программ [20 часов]**

Понятие функции. Спецификатор static. Фактические и формальные параметры функции. Способы передачи параметров в функцию (по значению, по ссылке, выходные параметры, параметры-массивы).

Лекции: 4 часа.

Практические занятия: 6 часов.

Самостоятельная работа: 10 часов.

#### **Литература по разделу:**

1. Канцелал С.А. Алгоритмизация и программирование: учебное пособие. – М.: ФОРУМ: НИЦ ИНФРА-М, 2014. – 352 с. – [Электронный ресурс]. – URL: <http://www.znaniyum.com>.
2. Гуриков С.Р. Введение в программирование на языке Visual C#: учебное пособие. – М.: Форум: НИЦ ИНФРА-М, 2013. – 448 с. – [Электронный ресурс]. – URL: <http://www.znaniyum.com>.
3. Подбельский В.В. Язык C#. Базовый курс: учеб.пос., 2-е изд., перераб.и доп.. – М.: Финансы и статистика; 2016. (Глава 1, Глава 2, Глава 3, Глава 4, Глава 6, Глава 9).
4. Павловская Т.А. C#. Программирование на языке высокого уровня. – СПб.: Питер, 2012. (Глава 1, Глава 2, Глава3, Глава 4, Глава 6).
5. Плаксин М.А. Тестирование и отладка программ – для профессионалов будущих и настоящих. М.: БИНОМ. Лаборатория базовых знаний, 2007.
6. Шилдт Г. C# 4.0. Полное руководство: пер. с англ. – М.: Издательский дом "Вильямс", 2015.

**Формы и методы проведения занятий по разделу, применяемые учебные технологии:** Практические занятия выполняются в форме лабораторных работ и предусматривают решение индивидуальных задач по программированию, для каждой задачи должен быть выполнен анализ, составлен алгоритм в виде блок-схемы, программа, набор тестов (критерии черного ящика, белого ящика, МГТ). Самостоятельная работа студентов предусматривает оформление отчета по лабораторной работе с описанием выполнения этапов жизненного цикла программного продукта.

## **Раздел II. Структурированные типы данных**

#### **Тема 6. Массивы [16 часов]**

Массивы в C#. Многомерные массивы. Рваные массивы. Цикл foreach для перебора массивов. Статические элементы класса Array для работы с массивами.



Типы задач по обработке массивов. Простые методы сортировки. Поисковые задачи. Двумерные массивы. Решение задач с использованием массивов.

Лекции: 2 часа.

Практические занятия: 4 часа.

Самостоятельная работа: 10 часов.

### **Тема 7. Строки [12 часов]**

Строковые литералы (регулярные и дословные). Тип string. Инициализация строк. Операции над строками. Статические элементы класса string обработки строковых данных. Форматирование строк. Массивы строк. Неизменяемость объектов типа string. Изменяемые строки (StringBuilder).

Лекции: 2 часа.

Практические занятия: 2 часа.

Самостоятельная работа: 8 часов.

### **Тема 8. Файлы [20 часов]**

Понятие потока. Классификация потоков. Основные классы для работы с файловыми потоками. Потоки байтов FileStream. Потоки символов (StreamWriter и StreamReader). Двоичные файлы (BinaryWriter, BinaryReader).

Лекции: 4 часа.

Практические занятия: 4 часа.

Самостоятельная работа: 12 часов.

### **Тема 9. Визуальное проектирование Windows приложений [22 часа]**

Основные особенности Windows. Типы программ в ОС Windows. Сообщения. Структура приложения с обработкой сообщений. Шаблон Windows – приложения. Основные типы Windows.Forms. Элементы управления. Диалоговые окна.

Лекции: 2 часа.

Практические занятия: 8 часов.

Самостоятельная работа: 12 часов.

### **Литература по разделу:**

1. Канцелл С.А. Алгоритмизация и программирование: учебное пособие. – М.: ФОРУМ: НИЦ ИНФРА-М, 2014. – 352 с. – [Электронный ресурс]. – URL: <http://www.znanium.com>.
2. Гуриков С.Р. Введение в программирование на языке Visual C#: учебное пособие. – М.: Форум: НИЦ ИНФРА-М, 2013. – 448 с. – [Электронный ресурс]. – URL: <http://www.znanium.com>.
3. Подбельский В.В. Язык C#. Базовый курс: учеб.пос., 2-е изд., перераб.и доп.. – М.: Финансы и статистика; 2016. (Глава 7, Глава 8)
4. Павловская Т.А. C#. Программирование на языке высокого уровня. – СПб.: Питер, 2012. (Глава 6, Глава 11, Глава 14)
5. Шилдт Г. C# 4.0. Полное руководство: пер. с англ. – М.: Издательский дом "Вильямс", 2015.

**Формы и методы проведения занятий по разделу, применяемые учебные технологии:** Практические занятия выполняются в форме лабораторных работ и предусматривают решение индивидуальных задач по программированию, для каждой задачи должен быть выполнен анализ, составлен алгоритм в виде блок-схемы, программа, набор тестов (критерии черного ящика, белого ящика, МГТ). Самостоятельная работа студентов предусматривает завершение работы над программой, начатой на практических занятиях в классе и оформление отчета по лабораторной работе с описанием выполнения этапов жизненного цикла программного продукта.



### Раздел III. Основные свойства ООП

#### Тема 10. Классы, объекты, методы. Перегрузка методов [22 часа]

Определение класса. Поля. Методы. Объекты. Конструкторы. Деструктор. Сбор мусора. Ключевое слово `this`. Ключевое слово `static`. Статические конструкторы. Статические классы. Перегрузка операций (унарные и бинарные операции, приведение типов).

Лекции: 2 часа.

Практические занятия: 6 часов.

Самостоятельная работа: 14 часов.

#### Тема 11. Основные свойства ООП: инкапсуляция [20 часов]

Инкапсуляция. Спецификаторы доступа. Свойства. Автоматические свойства. Индексаторы.

Лекции: 2 часа.

Практические занятия: 4 часа.

Самостоятельная работа: 14 часов.

#### Тема 12. Основные свойства ООП: наследование, полиморфизм [28 часов]

Иерархии классов. Наследование. Виртуальные методы. Механизм позднего связывания. Абстрактные классы. Класс `object`. Интерфейсы. Работа с объектами через стандартные интерфейсы `.NET`

Лекции: 4 часа.

Практические занятия: 6 часов.

Самостоятельная работа: 18 часов.

#### Литература по разделу:

1. Канцедаль С.А. Алгоритмизация и программирование: учебное пособие. – М.: ФОРУМ: НИЦ ИНФРА-М, 2014. – 352 с. – [Электронный ресурс]. – URL: <http://www.znaniyum.com>.
2. Гуриков С.Р. Введение в программирование на языке Visual C#: учебное пособие. – М.: Форум: НИЦ ИНФРА-М, 2013. – 448 с. – [Электронный ресурс]. – URL: <http://www.znaniyum.com>.
3. Подбельский В.В. Язык C#. Базовый курс: учеб.пос., 2-е изд., перераб.и доп.. – М.: Финансы и статистика; 2016. (Глава 10, Глава 11, Глава 12, Глава 13, Глава 14 )
4. Павловская Т.А. C#. Программирование на языке высокого уровня. – СПб.: Питер, 2012. (Глава 5, Глава 7, Глава 8, Глава 9)
5. Шилдт Г. C# 4.0. Полное руководство: пер. с англ. – М.: Издательский дом "Вильямс", 2015.

**Формы и методы проведения занятий по разделу, применяемые учебные технологии:** Практические занятия выполняются в форме лабораторных работ и предусматривают решение индивидуальных задач по программированию, для каждой задачи должен быть выполнен анализ, разработана диаграмма классов, программа, набор тестов (тестирование классов, тестирование программы). Самостоятельная работа студентов предусматривает завершение работы над программой, начатой на практических занятиях в классе.

### Раздел IV. Абстрактные типы данных. Обобщенное программирование.

#### Тема 13. Абстрактные типы данных. Коллекции [28 часов]

Основные абстрактные структуры данных. Физическое представление данных. Массив, список, двоичное дерево, хеш-таблица, стек, очередь Пространство имен `System.Collections`.



Интерфейсы. Коллекции общего назначения. Основные элементы класса ArrayList. Основные элементы класса Hashtable. Основные элементы класса SortedList. Основные элементы класса Stack. Основные элементы класса Queue. Доступ к коллекциям с помощью нумератора.

Лекции: 6 часов.

Практические занятия: 10 часов.

Самостоятельная работа: 12 часов.

#### **Тема 14. Обобщенное программирование [16 часов]**

Классы-прототипы. Основные преимущества использования обобщений. Ограничения. Параметризованные коллекции библиотеки .NET Создание собственных классов-прототипов. Обобщенные методы.

Лекции: 2 часа.

Практические занятия: 4 часа.

Самостоятельная работа: 10 часов.

#### **Литература по разделу:**

1. Канцелал С.А. Алгоритмизация и программирование: учебное пособие. – М.: ФОРУМ: НИЦ ИНФРА-М, 2014. – 352 с. – [Электронный ресурс]. – URL: <http://www.znanium.com>.
2. Гуриков С.Р. Введение в программирование на языке Visual C#: учебное пособие. – М.: Форум: НИЦ ИНФРА-М, 2013. – 448 с. – [Электронный ресурс]. – URL: <http://www.znanium.com>.
3. Подбельский В.В. Язык C#. Базовый курс: учеб.пос., 2-е изд., перераб.и доп.. – М.: Финансы и статистика; 2016. (Глава 18).
4. Павловская Т.А. C#. Программирование на языке высокого уровня. – СПб.: Питер, 2012. (Глава 13)
5. Плаксин М.А. Тестирование и отладка программ – для профессионалов будущих и настоящих. М.: БИНОМ. Лаборатория базовых знаний, 2007.
6. Шилдт Г. C# 4.0. Полное руководство: пер. с англ. – М.: Издательский дом "Вильямс", 2015.

**Формы и методы проведения занятий по разделу, применяемые учебные технологии:** Практические занятия выполняются в форме лабораторных работ и предусматривают решение индивидуальных задач по программированию, для каждой задачи должен быть выполнен анализ, разработана диаграмма классов, программа, набор тестов (тестирование классов, тестирование программы). Самостоятельная работа студентов предусматривает завершение работы над программой, начатой на практических занятиях в классе.

#### **Раздел V. Объектно-событийное и объектно-ориентированное программирование.**

##### **Тема 15. Обработка исключительных ситуаций [16 часов]**

Механизм обработки исключений. Системные исключения и их обработка. Свойства и методы класса Exception. Исключения, создаваемые программистом.

Лекции: 4 часа.

Практические занятия: 4 часа.

Самостоятельная работа: 8 часов.

##### **Тема 15. ОО программа как программа, управляемая событиями [28 часов]**





Понятие делегата. Описание и использование. Многоадресная передача. События. События для многоадресной передачи. Многопоточные приложения.

Лекции: 6 часов.

Практические занятия: 10 часов.

Самостоятельная работа: 10 часов.

#### **Литература по разделу:**

1. Канцедал С.А. Алгоритмизация и программирование: учебное пособие. – М.: ФОРУМ: НИЦ ИНФРА-М, 2014. – 352 с. – [Электронный ресурс]. – URL: <http://www.znanium.com>.
2. Гуриков С.Р. Введение в программирование на языке Visual C#: учебное пособие. – М.: Форум: НИЦ ИНФРА-М, 2013. – 448 с. – [Электронный ресурс]. – URL: <http://www.znanium.com>.
3. Подбельский В.В. Язык C#. Базовый курс: учеб.пос., 2-е изд., перераб.и доп.. – М.: Финансы и статистика; 2016. (Глава 16, Глава 17)
4. Павловская Т.А. C#. Программирование на языке высокого уровня. – СПб.: Питер, 2012. (Глава 4, Глава 10)
5. Шилдт Г. C# 4.0. Полное руководство: пер. с англ. – М.: Издательский дом "Вильямс", 2015.

**Формы и методы проведения занятий по разделу, применяемые учебные технологии:** Практические занятия выполняются в форме лабораторных работ и предусматривают решение индивидуальных задач по программированию, для каждой задачи должен быть выполнен анализ, разработана диаграмма классов, программа, набор тестов (тестирование классов, тестирование программы). Самостоятельная работа студентов предусматривает завершение работы над программой, начатой на практических занятиях в классе.

## **Раздел VI. Запросы LINQ to objects**

### **Тема 17. Дополнительные возможности [18 часов]**

Неявная типизация локальных переменных. Синтаксис инициализации объектов и коллекций. Лямбда-выражения. Расширяющие методы. Анонимные типы.

Лекции: 4 часа.

Практические занятия: 4 часа.

Самостоятельная работа: 10 часов.

### **Тема 18. LINQ to objects [18 часов]**

Применение запросов LINQ к элементарным массивам. Возврат результатов LINQ запросов. Применение запросов LINQ к объектам коллекций. Применение запросов LINQ к необобщенным коллекциям. Внутреннее представление операторов запросов LINQ.

Лекции: 4 часа.

Практические занятия: 4 часа.

Самостоятельная работа: 10 часов.

#### **Литература по разделу:**

1. Канцедал С.А. Алгоритмизация и программирование: учебное пособие. – М.: ФОРУМ: НИЦ ИНФРА-М, 2014. – 352 с. – [Электронный ресурс]. – URL: <http://www.znanium.com>.
2. Гуриков С.Р. Введение в программирование на языке Visual C#: учебное пособие. – М.: Форум: НИЦ ИНФРА-М, 2013. – 448 с. – [Электронный ресурс]. – URL: <http://www.znanium.com>.
3. Подбельский В.В. Язык C#. Базовый курс: учеб.пос., 2-е изд., перераб.и доп.. – М.: Финансы и статистика; 2016.





4. Павловская Т.А. С#. Программирование на языке высокого уровня. – СПб.: Питер, 2012.
5. Шилдт Г. С# 4.0. Полное руководство: пер. с англ. – М.: Издательский дом "Вильямс", 2015.
6. Троелсен, Эндрю. Язык программирования С# 5.0 и платформа .NET 4.5, 6-е изд. : Пер. с англ. —М. : ООО “И.Д. Вильямс”, 2013. — 1312 с.

**Формы и методы проведения занятий по разделу, применяемые учебные технологии:** Практические занятия выполняются в форме лабораторных работ и предусматривают решение индивидуальных задач по программированию, для каждой задачи должен быть выполнен анализ, разработана диаграмма классов, программа, набор тестов (тестирование классов, тестирование программы). Самостоятельная работа студентов предусматривает завершение работы над программой, начатой на практических занятиях в классе.

## **Раздел VII. Многопоточное программирование**

### **Тема 19. Создание потоков [20 часов]**

Введение в многопоточность. Класс Thread. Создание потоков. Потоки с параметрами.

Лекции: 6 часов.

Практические занятия: 6 часов.

Самостоятельная работа: 8 часов.

### **Тема 20. Синхронизация потоков [22 часа]**

Проблема синхронизации потоков. Мониторы. Мьютексы. Семафоры. Использование таймеров.

Лекции: 6 часов.

Практические занятия: 6 часов.

Самостоятельная работа: 10 часов.

### **Литература по разделу:**

7. Канцедаль С.А. Алгоритмизация и программирование: учебное пособие. – М.: ФОРУМ: НИЦ ИНФРА-М, 2014. – 352 с. – [Электронный ресурс]. – URL: <http://www.znaniyum.com>.
8. Гуриков С.Р. Введение в программирование на языке Visual C#: учебное пособие. – М.: Форум: НИЦ ИНФРА-М, 2013. – 448 с. – [Электронный ресурс]. – URL: <http://www.znaniyum.com>.
9. Подбельский В.В. Язык С#. Базовый курс: учеб.пос., 2-е изд., перераб.и доп.. – М.: Финансы и статистика; 2016.
10. Павловская Т.А. С#. Программирование на языке высокого уровня. – СПб.: Питер, 2012.
11. Шилдт Г. С# 4.0. Полное руководство: пер. с англ. – М.: Издательский дом "Вильямс", 2015.
12. Троелсен, Эндрю. Язык программирования С# 5.0 и платформа .NET 4.5, 6-е изд. : Пер. с англ. —М. : ООО “И.Д. Вильямс”, 2013. — 1312 с.

**Формы и методы проведения занятий по разделу, применяемые учебные технологии:** Практические занятия выполняются в форме лабораторных работ и предусматривают решение индивидуальных задач по программированию, для каждой задачи должен быть выполнен анализ, разработана диаграмма классов, программа, набор тестов (тестирование классов, тестирование программы). Самостоятельная работа студентов предусматривает завершение работы над программой, начатой на практических занятиях в классе.



## 9. Образовательные технологии

### 9.1. Методические рекомендации преподавателю

На лекциях используется «проблемный» подход к изложению материала: материал каждой лекции иллюстрируется примерами, рассматриваются нестандартные ситуации, требующие решения с использованием рассматриваемого материала. При этом студенты должны активно участвовать в обсуждении вопросов, выработке решений. Для самостоятельного изучения предлагается использовать электронные ресурсы.

На практических занятиях используются следующие методы обучения и контроля усвоения материала:

- выполнение практических работ предполагает решение индивидуальных задач по программированию в форме лабораторных работ, по работе оформляется отчет, описывающий процесс решения задачи в соответствии с жизненным циклом программной системы;
- каждую лабораторную работу студент защищает преподавателю и получает оценку за защиту, в рамках защиты обсуждаются различные варианты решения, предложенные студентами, сравнение решений, анализ возможных ситуаций, code review.

### 9.2. Методические указания студентам

Студенту рекомендуется следующая схема подготовки к практическому занятию:

- 1) проработать конспект лекций;
- 2) проанализировать основную и дополнительную литературу, рекомендованную по изучаемому разделу;
- 3) проанализировать варианты решений, предложенные преподавателем;
- 4) при затруднениях сформулировать вопросы к преподавателю.

Для подготовки к занятиям рекомендуется использовать источники, представленные в электронных форматах: рассмотреть примеры решений, использовать пошаговое выполнение программ в среде разработки.

## 10. Оценочные средства для текущего контроля и аттестации студента

### 10.1. Оценочные средства для оценки качества освоения дисциплины в ходе текущего контроля

Примерные вопросы/ задания для домашнего задания:

#### Домашнее задание №1

Разработка консольного приложения для обработки одномерных, двумерных, рваных массивов и строк (string). Задача состоит в формировании указанной структуры данных и ее обработки в соответствии с вариантом (удаление элементов /строк /столбцов/слов, добавлении элементов /строк /столбцов/ слов). Предусмотреть запись элементов структуры в файл и загрузку структуры из файла. Должен быть организован интерфейс пользователя в виде меню и предусмотрена обработка ошибок при неправильном вводе данных пользователем.

#### Домашнее задание №2

Разработка Windows-приложения, выполняющего следующие функции:

1. Создает объекты из иерархии классов заданной предметной области.
2. Записывает объекты в коллекцию (массив).
3. Выводит объекты для просмотра коллекции.
4. Удаляет и добавляет объекты в коллекцию.
5. Выполняет запросы, поиск заданных элементов.
6. Сортирует коллекцию.



7. Очищает коллекцию.

Вопросы для оценки качества освоения дисциплины:

1. Жизненный цикл программного обеспечения.
2. Каскадная схема жизненного цикла ПО.
3. Итерационная схема жизненного цикла ПО.
4. Основные участники разработки ПО.
5. Основные парадигмы программирования.
6. Процедурно-ориентированная парадигма программирования.
7. Объектно-ориентированная парадигма программирования.
8. Цель тестирования. Принципы тестирования. Полнота тестирования
9. Отличия критериев Черного ящика и Белого ящика.
10. Критерии Черного ящика.
11. Тестирование упорядоченности
12. Критерии белого ящика.
13. Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма.
14. Понятие системы программирования. Система программирования VS.Net.
15. Общая характеристика платформы MSDN.
16. Структура программы на языке программирования C#.
17. Основные элементы языка программирования.
18. Понятие переменной.
19. Понятие типа данных.
20. Классификация типов данных.
21. Встроенные типы данных.
22. Область действия идентификатора.
23. Локальные и глобальные переменные.
24. Основные операции.
25. Преобразование типов.
26. Понятие исключительной ситуации.
27. Чтение данных с клавиатуры.
28. Вывод данных на экран.
29. Оператор присваивания
30. Составной оператор.
31. Условный оператор.
32. Оператор выбора.
33. Цикл с предусловием.
34. Цикл с постусловием.
35. Цикл со счетчиком.
36. Массивы.
37. Оператор foreach.
38. Размерность массива. Многомерные массивы.
39. Рваные массивы.
40. Типы задач по обработке массивов.
41. Бинарный поиск в массиве
42. Сортировка простым обменом
43. Сортировка простым выбором
44. Сортировка простым включением.
45. Методы и свойства класса Array.
46. Регулярные и буквальное строковые литералы.
47. Ссылки типа string.
48. Операции над строками
49. Методы и свойства класса String.



50. Форматирование строк.
51. Неизменяемость объектов класса String.
52. Тип System.Text.StringBuilder.
53. Функции.
54. Параметры функции.
55. Переменным числом параметров.
56. Функции с необязательными параметрами.
57. Функции с именованными параметрами.
58. Перегрузка методов.
59. Рекурсия.
60. Использование массивов в качестве аргументов и возвращаемых значений функций.
61. Понятие потока. Классификация потоков.
62. Основные классы для работы с файловыми потоками.
63. Режимы доступа к файлу, режимы открытия файла, режимы совместного использования файла.
64. Потоки байтов FileStream.
65. Потоки символов StreamWriter и StreamReader.
66. Двоичные файлы BinaryWriter, BinaryReader.
67. Особенности приложения под ОС Windows.
68. Структура приложения Microsoft Windows.
69. Процесс обработки сообщений в приложениях под ОС Windows. Цикл обработки сообщений.
70. Создание приложений под ОС Windows.
71. Управляющие элементы в приложениях под ОС Windows.
72. Определение класса.
73. Поля. Методы.
74. Объекты.
75. Конструкторы.
76. Деструктор. Сбор мусора.
77. Ключевое слово this. Ключевое слово static. Статические конструкторы. Статические классы.
78. Перегрузка операций (унарные и бинарные операции, приведение типов).
79. Основные свойства ООП.
80. Инкапсуляция. Спецификаторы доступа.
81. Свойства. Автоматические свойства.
82. Индексаторы.
83. Иерархии классов.
84. Наследование.
85. Виртуальные методы. Механизм позднего связывания.
86. Абстрактные классы.
87. Класс object.
88. Интерфейсы.
89. Работа с объектами через стандартные интерфейсы .NET
90. Понятие делегата. Описание и использование.
91. Многоадресная передача.
92. События.
93. События для многоадресной передачи.
94. Многопоточные приложения.
95. Механизм обработки исключений.
96. Системные исключения и их обработка.
97. Свойства и методы класса Exception.

98. Исключения, создаваемые программистом.
99. Основные абстрактные структуры данных.
100. Физическое представление данных.
101. Пространство имен System.Collections.
102. Интерфейсы пространства имен System.Collections.
103. Коллекции общего назначения.
104. Основные элементы класса ArrayList.
105. Основные элементы класса Hashtable.
106. Основные элементы класса SortedList.
107. Основные элементы класса Stack.
108. Основные элементы класса Queue.
109. Доступ к коллекциям с помощью нумератора.
110. Классы-прототипы.
111. Основные преимущества использования обобщений.
112. Ограничения.
113. Параметризованные коллекции библиотеки .NET.
114. Создание собственных классов-прототипов.
115. Обобщенные методы.

## 10.2. Примеры заданий промежуточной аттестации

1. Описать иерархию классов: **Pair**( int first, int second) – базовый класс, **Time** (минуты, секунды) – производный класс. Для каждого класса определить данные, свойства, конструкторы без параметров, с параметрами, метод для инициализации объектов, метод для вывода данных объекта.
2. В основной программе создать массив, в который записать объекты разных классов (базового и производного).
3. Подсчитать количество объектов каждого класса в массиве.
4. Выполнить сортировку объектов в массиве по возрастанию.
5. Выполнить клонирование объектов типа Time.

## 11. Порядок формирования оценок по дисциплине

Преподаватель на практических занятиях, проводимых в компьютерных классах, оценивает работу студентов. Программы, написанные во время практических занятий, должны соответствовать следующим критериям: отсутствие синтаксических ошибок, соответствие спецификации (реализованы все основные функции), реализованы дополнительные функции, программа правильно работает для разных наборов исходных данных, разработаны тесты с проверкой достаточности по критериям черного ящика и МГТ, стиль программы (комментарии, отступы, правильные имена переменных), модульность (использование классов/функций), удобный интерфейс, студент может объяснить работу программы, знает принципы и механизмы, используемые в работе программы, программа должна быть выполнена и защищена в сроки, указанные преподавателем, при задержке сдачи работы оценка может быть снижена. Оценки за работу на практических занятиях преподаватель выставляет в рабочую ведомость. Оценка по 10-ти балльной шкале за работу на практических занятиях определяется перед промежуточным или итоговым контролем - *О аудиторная*.

Самостоятельная работа оценивается по результатам оформления отчета, отчет должен быть загружен в LMS в сроки, указанные преподавателем, при задержке сдачи работы оценка может быть снижена.



Оценки за самостоятельную работу студента преподаватель выставляет в рабочую ведомость. Оценка по 10-ти балльной шкале за самостоятельную работу определяется перед промежуточным или завершающим контролем -  $O_{сам}$ .

**Накопленная оценка по дисциплине** ( $O_{накопленная}$ ) рассчитывается по формуле:

$$O_{накопленная} = k_1 * O_{текущий} + k_2 * O_{ауд} + k_3 * O_{сам. работа}$$

где  $k_1=0,6$ ;  $k_2=0,2$ ,  $k_3=0,2$ .

Способ округления накопленной по дисциплине оценки: арифметический.

**Оценка за текущий контроль** ( $O_{текущий}$ ) рассчитывается как взвешенная сумма всех форм текущего контроля.

$$O_{текущий} = n_1 * O_{дз} + n_2 * O_{дз},$$

где  $n_1=0,5$ ;  $n_2=0,5$

Способ округления оценки за текущий контроль: арифметический.

**Результирующая оценка** за дисциплину рассчитывается следующим образом:

$$O_{результ} = k_1 * O_{накопл} + k_2 * O_{экз}$$

где  $k_1=0,6$ ;  $k_2=0,4$ .

Способ округления результирующей по дисциплине оценки: арифметический.

На экзамене студент может получить дополнительный вопрос (дополнительную практическую задачу, решить к передаче домашнее задание), ответ на который оценивается в 1 балл.

## 12. Учебно-методическое и информационное обеспечение дисциплины

### 12.1. Основная литература

1. Канцедаль С.А. Алгоритмизация и программирование: учебное пособие. – М.: ФОРУМ: НИЦ ИНФРА-М, 2014. – 352 с. – [Электронный ресурс]. – URL: <http://www.znaniyum.com>.
2. Гуриков С.Р. Введение в программирование на языке Visual C#: учебное пособие. – М.: Форум: НИЦ ИНФРА-М, 2013. – 448 с. – [Электронный ресурс]. – URL: <http://www.znaniyum.com>.

### 12.2. Дополнительная литература

3. Подбельский В.В. Язык C#. Базовый курс: учеб.пос., 2-е изд., перераб.и доп.. – М.: Финансы и статистика; 2016.
4. Павловская Т.А. C#. Программирование на языке высокого уровня. – СПб.: Питер, 2012.
5. Шилдт Г. C# 4.0. Полное руководство: пер. с англ. – М.: Издательский дом "Вильямс", 2015.
6. Троелсен, Эндрю. Язык программирования C# 5.0 и платформа .NET 4.5, 6-е изд. : Пер. с англ. —М. : ООО "И.Д. Вильямс", 2013. — 1312 с.

### 12.3. Справочники, словари, энциклопедии

Интерактивная справочная система MSDN.

### 12.4. Программные средства

Для успешного освоения дисциплины, студент использует следующие программные средства:



- MS Visual Studio 2015/2017.
- Интегрированный пакет Microsoft Office (MS Word и Excel) для оформления тестов и тестирования программ, а также отчетов по выполненным заданиям.
- Средства, обеспечивающие возможность доступа к материалам для подготовки к занятиям в различных форматах (документы MS Word, документы в формате HTML, презентации MS Power Point), размещенные на сервере, доступные в Internet.

#### **12.5. Дистанционная поддержка дисциплины**

Дистанционная поддержка курса предусмотрена в рамках информационной образовательной среды LMS.

#### **13. Материально-техническое обеспечение дисциплины**

Для проведения лекционных занятий используется компьютер с установленным программным обеспечением для демонстрации презентаций и проектор.

Практические занятия проводятся в компьютерных классах с установленным программным обеспечением, перечисленным выше.