

Ульяновский государственный университет

Институт экономики и бизнеса

Кафедра цифровой экономики

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ  
ПО САМОСТОЯТЕЛЬНОЙ РАБОТЕ  
И ИЗУЧЕНИЮ ДИСЦИПЛИНЫ  
«Программирование»**

для направления подготовки

38.03.05 Бизнес-информатика, профиль «Цифровая экономика»

Ульяновск – 2018

Методические рекомендации по самостоятельной работе и изучению дисциплины «**Программирование**» для направления подготовки 38.03.05 Бизнес-информатика, профиль «Цифровая экономика» / Составитель Мартыненко Ю.В. УлГУ, Институт Экономики и Бизнеса. - Ульяновск, 2018. – 21 с.

Рекомендовано к введению в образовательный процесс решением Ученого совета Института экономики и бизнеса УлГУ от 24 мая 2018 г., протокол № 213/09.

Настоящие методические рекомендации предназначены для студентов направления подготовки 38.03.05 Бизнес-информатика, профиль «Цифровая экономика»

Составитель: к.ф.-м.н., доцент кафедры цифровой экономики Мартыненко Ю.В.

## МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина **«Программирование»** принадлежит базовой части ОПОП ВО по направлению «Бизнес-информатика». Дисциплина изучается студентами 1 курса во 2 семестре.

Изучение дисциплины «Программирование» базируется на знаниях, умениях и компетенциях, сформированных у обучающихся в процессе изучения дисциплин Введение в специальность, Информационные технологии в экономике и управлении, Информатика.

Параллельно с овладением знаниями, умениями и навыками, полученными в результате изучения дисциплины «Программирование» осваиваются знания, умения и навыки, получаемые в результате изучения дисциплины Теоретические основы информатики.

Знания, умения и навыки, полученные при изучении дисциплины «Программирование», необходимы для изучения таких дисциплин Проектирование систем ключевых показателей предприятия, Инструменты цифровой экономики, Моделирование бизнес-процессов, Имитационное моделирование, Базы данных, Распределенные системы в цифровой экономике, Информационная безопасность, Технология блок-чейн и криптовалюта, Управление ИТ сервисами и контентом, Системы управления контентом, а также прохождения практик Практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности, «Проектная деятельность», Научно-исследовательская работа, Преддипломная практика; в процессе прохождения государственной итоговой аттестации: при подготовке к сдаче и сдаче государственного экзамена, защите выпускной квалификационной работы, включая подготовку к процедуре защиты и процедуру защиты.

**ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО  
ДИСЦИПЛИНЕ(МОДУЛЮ), СООТНЕСЕННЫХ С ПЛАНИРУЕМЫМИ  
РЕЗУЛЬТАТАМИ ОСВОЕНИЯ ОСНОВНОЙ ПРОФЕССИОНАЛЬНОЙ  
ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

В результате освоения дисциплины формируются следующие компетенции:

Код и наименование реализуемой компетенции	Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций
<p>ОПК-3 способность работать с компьютером как средством управления информацией, работать с информацией из различных источников, в том числе в глобальных компьютерных сетях</p>	<p>Знать: современные методы и средства цифровой обработки информации.</p> <p>Уметь: работать в информационно-коммуникационном пространстве, решать профессиональные задачи с использованием программных средств общего и специального назначения.</p> <p>Владеть: навыками подбора оптимальных программных средств для решения поставленной задачи.</p>
<p>ПК-13 умение проектировать и внедрять компоненты ИТ-инфраструктуры предприятия, обеспечивающие достижение стратегических целей и поддержку бизнес-процессов</p>	<p>Знать: возможности типовой ИС, устройство и функционирование современных ИС, основы программирования, программные средства и платформы разработки ИС.</p> <p>Уметь: проводить анализ предметной области, выявлять границы проведения работ по внедрению ИС, формулировать ТЗ на разработку ИС.</p> <p>Владеть: навыками функциональной декомпозиции задачи, подбора методов и алгоритмов ее решения.</p>

ПК-16 умение разрабатывать контент и ИТ-сервисы предприятия и интернет- ресурсов	Знать: языки программирования, методы и средства тестирования программ, методы прототипирования пользовательского интерфейса. Уметь: кодировать на языках программирования, тестировать разработанный код. Владеть: способностью использовать языки программирования для решения профессиональных задач.
--	--

### ОБЩАЯ ТРУДОЕМКОСТЬ ДИСЦИПЛИНЫ

**Объем дисциплины в зачетных единицах (всего:) 5 зачетных единиц**

**Объем дисциплины по видам учебной работы (в часах)**

Вид учебной работы	Количество часов (форма обучения очная)			
	Всего по плану	В т.ч. по семестрам		
		1	2	3
1	2	3	4	5
Контактная работа обучающихся с преподавателем в соответствии с УП	64	64		
Аудиторные занятия:	64	64		
Лекции	32	32		
Семинары и практические занятия	0	0		
Лабораторные работы, практикумы	32	32		



1	2	3	4	5	6	7	8
Алгоритм и алгоритмические структуры. Теоретические основы программирования	24	4	0	0	0	20	эссе
Основы языка Delphi	75	18	0	27	27	30	Тестирование, контрольная работа, опрос
Основы объектно-ориентированного программирования	45	10	0	5	5	30	Контрольная работа, опрос
Подготовка и сдача курсовой работы, экзамена	36					36	
Итого	180	32	0	32	32	116	

## ОРГАНИЗАЦИЯ ИЗУЧЕНИЯ ДИСЦИПЛИНЫ

Содержание дисциплины «**Программирование**» разбито на темы с определением количества как аудиторной, так и самостоятельной работы. Предусмотрены следующие виды аудиторной работы: лекции, лабораторные работы. Лекции составляют основу теоретической подготовки в рамках дисциплины «**Программирование**» и предусматривают преимущественную передачу учебной информации преподавателем обучающимся. Они дают систематизированные основы знаний по дисциплине, раскрывают основные приемы программирования на языке высокого уровня.

Специфика дисциплины «**Программирование**» обуславливает проведение следующих типов лекций:

- лекция-информация, ориентированная на изложение и объяснение информации, подлежащей осмыслению и запоминанию обучающимся;
- проблемная лекция, где новое знание вводится через практическую задачу в области программирования.

Во время лекции обучающемуся рекомендуется писать ее конспект.

Написание конспекта лекции формирует навыки выделения наиболее важных элементов изучаемого материала. Кроме этого, конспект позволяет зафиксировать в ходе проводимого занятия возникающие вопросы, подлежащие дальнейшему изучению за пределами лекционных занятий.

Лабораторные работы являются одним из видов практических занятий, проводимых с применением современных информационных технологий, на которых предполагается в значительной степени самостоятельное выполнение обучающимся задания, направленного на закрепление и углубление знаний, полученных на лекционных занятиях. Методические указания к выполнению лабораторных работ представлены в приложении к РПД «**Программирование**».

Также по каждой теме дисциплины предусмотрен необходимый объем самостоятельной работы. Самостоятельная работа обучающегося – это способ активного, целенаправленного приобретения обучающимся новых для него знаний, умений и навыков с участием и без участия в этом процессе преподавателя. Самостоятельная работа проводится с целями: систематизация и закрепление полученных теоретических знаний и практических умений обучающихся; углубления и расширения теоретических знаний; формирования умений использовать нормативную, справочную и иную необходимую документацию и специальную литературу; приобретения навыков решения профессиональных задач.

Рекомендуется, что необходимый объем самостоятельной работы по теме осваивается обучающимся через текущую самостоятельную работу и внеаудиторную самостоятельную работу.

Текущая самостоятельная работа обеспечивает подготовку к текущим аудиторным занятиям и контрольным мероприятиям в рамках изучения дисциплины. Результаты этой подготовки проявляются в активности студента на занятиях и в качестве выполненных лабораторных работ.

Внеаудиторная самостоятельная работа – это часть учебного процесса, выполняемая обучающимся во внеаудиторное время по заданию и при



методическом руководстве преподавателя, но без его непосредственного участия. Она направлена на углубление и закрепление знаний обучающегося, полученных им на аудиторных занятиях, формирование и развитие соответствующих компетенций, умений и навыков. Дисциплина «**Программирование**» предусматривает самостоятельную работу в виде написания эссе, подготовки к тестированию, написания двух контрольных работ, подготовки к сдаче лабораторных работ. Также эта форма самостоятельной работы включает в себя подготовку к экзамену по дисциплине.

## **РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К АУДИТОРНЫМ ЗАНЯТИЯМ**

Подготовка к лекциям включает в себя проработку конспекта лекций и самостоятельное изучение литературы. Необходимость самостоятельной работы по подготовке к лекциям определяется тем, что изучение дисциплины строится по определенной логике освоения ее тем, представленных в рабочей программе дисциплины. Преподаватель при чтении новой лекции указывает на связь ее содержания с тем, которое было прежде изучено. Поэтому, если предыдущие лекции были пропущены, возникают сложности с освоением нового материала. Качество освоения содержания дисциплины прямо зависит оттого, насколько обучающийся сам, без внешнего принуждения формирует у себя установку на получение на лекциях новых знаний, дополняющих уже имеющиеся по данной дисциплине.

Рекомендуется следующая схема работы. Перед лекцией следует ознакомиться с соответствующим материалом из учебной литературы. Это позволит лучше усвоить содержание лекции. Во время лекции составляется ее конспект, в котором зафиксированы основные положения лекции, а также изученные примеры, комментарии к ним, и иная информация, которую обучающийся считает нужным зафиксировать. После прохождения лекции

рекомендуется повторно обратиться к учебной литературе, сопоставить полученную информацию, и, при необходимости, дополнить конспект. Также следует составить список вопросов, оставшихся непонятыми, чтобы на следующем занятии обратиться к преподавателю за разъяснением.

### **Список рекомендованного для изучения материала**

#### **Тема 1. Алгоритм и алгоритмические структуры. Теоретические основы программирования - 2 лекции**

##### **Лекция 1. Алгоритмы и языки программирования**

[1] – лекция 1

[4] – глава 1, глава 2

##### **Лекция 2. Вопросы разработки ПО**

[3] – главы 3,6

#### **Тема 2. Основы языка Delphi – 9 лекций**

##### **Лекция 1. Переменные, оператор присваивания, выражения**

[1] – Лекция 2

[2] – Лекция 4

##### **Лекция 2. Условный оператор**

[1] – Лекция 4

[2] – Лекция 7

##### **Лекция 3. Оператор цикла**

[1] – Лекция 4

[2] – Лекция 10

##### **Лекция 4. Процедуры и функции**

[1] – Лекция 3

[2] – Лекция 9

##### **Лекция 5. Работа со строками и символами**

[1] – Лекция 10

[2] – Лекция 5-6

## **Лекция 6. Работа с массивами часть 1**

[1] – Лекция 9

[2] – Лекция 13

## **Лекция 7. Работа с массивами часть 2**

[1] – Лекция 22

[2] – Лекция 14

## **Лекция 8. Работа с текстовыми файлами**

[1] – Лекция 19

[2] – Лекция 21

## **Лекция 9. Работа с файлами**

[1] – Лекция 20

[2] – Лекция 21

## **Тема 3. Основы объектно-ориентированного программирования – 5 лекций**

### **Лекция 1. Концепция объектно-ориентированного программирования**

[1] – Лекция 6

[2] – Лекция 3

[3] – Глава 6

### **Лекция 2. Работа с классами: TMemo, TEdit, TMenu**

[1] – Лекция 8

[2] – Лекция 6

### **Лекция 3. Работа с компонентами Dialog**

[1] – Лекция 13

[2] – Лекция 14

### **Лекция 4. Работа со списками**

[1] – Лекция 12

[2] – Лекция 13

### **Лекция 5. Работа с паттернами**

[3] – глава 7

Подготовка к лабораторным работам включает в себя повторение необходимого для выполнения работы теоретического материала. На лабораторном занятии следует внимательно ознакомиться с предложенным заданием, выполнить его, консультируясь по мере надобности с преподавателем, а затем сдать задание преподавателю. Преподаватель может задать уточняющие вопросы или попросить что-либо скорректировать в выполненной работе.

После проведения лабораторной работы следует проанализировать полученный результат, замечания и комментарии преподавателя о выполненной работе. Рекомендуется составить в свободной форме конспект выполнения работы, зафиксировав в нем наиболее важные и сложные моменты, в том числе рефлексию своих образовательных результатов. В таком случае обучающийся фиксирует, что он знал и умел в начале проекта, какие навыки и знания он получил в процессе работы.

Данный конспект может быть использован и при подготовке к следующим лабораторным работам и/или другим формам работы в ходе изучения дисциплины.

При согласовании с преподавателем задание на лабораторную работу может быть групповым, т.е. выполняться совместно несколькими обучающимися. В этом случае дополнительно рекомендуется провести совместное обсуждение выполненной работы внутри своей группы.

## **Список вопросов для подготовки к сдаче лабораторных работ**

### **Тема 2. Основы языка Delphi**

1) Какие базовые алгоритмические структуры использованы при выполнении работы?

- 2) Какие операторы использованы при выполнении работы?
- 3) Какие компоненты использованы при выполнении работы?
- 4) Каким образом программа получает входные данные?
- 5) Каким образом программа обрабатывает исключительные ситуации?

### **Тема 3. Основы объектно-ориентированного программирования**

- 1) Какие элементы концепции ООП использованы в программе?
- 2) Какие элементы класса использованы в программе?
- 3) Как создаются на основе класса объекты?
- 4) Как обрабатываются исключительные ситуации?
- 5) Как объекты получают данные для работы?

## **РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ЭКЗАМЕНУ**

Изучение дисциплины **«Программирование»** завершается промежуточной аттестацией в форме экзамена. Экзамен представляет собой форму контроля учебной деятельности студента, которая используется, если объем учебной дисциплины составляет две и более зачетных единиц, т. е. более 72 часов. Оценка выявленных на экзамене знаний, умений и компетенций дифференцирована: в зачетной книжке ставится оценка «удовлетворительно», «хорошо» или «отлично».

Самостоятельная подготовка к экзамену должна осуществляться в течение всего семестра, а не за несколько дней до его проведения. Подготовка включает следующие действия. Прежде всего нужно перечитать все лекции, а также иные материалы, подготовленные в рамках изучения дисциплины.

Затем надо соотнести эту информацию с вопросами, которые даны к экзамену. Если информации недостаточно, ответы находят в предложенной преподавателем литературе. Важно сформировать целостное представление о содержании ответа на каждый вопрос, что предполагает знание разных

трактовок сущности того или иного явления, процесса, умение раскрывать факторы, определяющие их взаимоотношения.

Рекомендуется подготовку к экзамену осуществлять в два этапа. На первом, в течение 2–3 дней, подбирается из разных источников весь материал, необходимый для развернутых ответов на все вопросы. Ответы можно записать в виде краткого конспекта. На втором этапе по памяти восстанавливается содержание того, что записано в ответах на каждый вопрос. Если какие-либо вопросы остаются непроясненными, их следует задать преподавателю на консультации перед экзаменом. Накануне экзамена необходимо повторить ответы, не заглядывая в записи.

### **Список вопросов для подготовки**

1. Алгоритмические языки программирования.
2. Языки программирования высокого уровня.
3. «Жизненный цикл» программы.
4. Среда Delphi, назначение ее компонент.
5. Главное окно, окно формы, окно инспектора объектов, окно кода программы.
6. Структура программ Delphi.
7. Элементы языка Delphi.
8. Операторы языка Delphi.
9. Простые и структурированные типы данных.
10. Строки.
11. Файлы.
12. Указатели и динамическая память.
13. Подпрограммы и их описания.
14. Процедуры.
15. Функции.
16. Модули.
17. Классы.

18. Объекты.
19. Основные компоненты визуального программирования.

### **РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К НАПИСАНИЮ ЭССЕ**

Текущий контроль по дисциплине «**Программирование**» предусматривает написание эссе, направленного на проверку уровня усвоения материала темы 1. Эссе – это самостоятельная письменная работа на предложенную тему, целью которой является развитие навыков самостоятельного творческого мышления и письменного изложения мыслей обучающегося. Эссе выражает индивидуальное мнение и размышления обучающегося на предложенную тему в области программирования.

Требуемый объем эссе – 4-5 страниц, оформление по правилам: формат А4, через полтора интервала, шрифтом Times New Roman стандартного размера (14-й шрифт компьютера), с соблюдением размеров отступа от края листа: левое поле – 30 мм, правое – 10 мм, верхнее и нижнее поля – по 20 мм. Текст должен быть форматирован «по ширине» (т.е. выровнен по вертикали, как с левой, так и с правой стороны листа). Обязательно следует соблюдать абзацный отступ размером 1,25 см (5 знаков). Цвет шрифта печати основного текста должен быть черным, использование иных цветов возможно только для оформления специальных схем, чертежей, рисунков.

Содержание и титульный лист не требуются, на первом листе в левом верхнем углу указываются ФИО обучающегося и группа, на следующей строке по центру указывается тема, далее идет основная часть. Список источников не указывается, однако обучающийся должен быть готов ответить на вопросы преподавателя о том, какие источники он использовал в своей работе.

Композиция эссе свободная, исходя из темы и авторского видения обучающегося. Обязательным требованием является выражение собственной

точки зрения на изучаемую проблему. Рекомендуется следующий порядок написания эссе:

1. внимательно прочитать тему, подобрать и изучить источники по теме, содержащуюся в них информацию;
2. выбрать имеющуюся в теме актуальную проблему, если необходимо, собрать по ней дополнительный материал;
3. составить план эссе;
4. лаконично, но емко раскрыть содержание проблемы и свои подходы к ее решению;
5. оформить эссе и сдать в установленный срок.

Критерием оценки эссе является правильность и полнота раскрытия выбранных вопросов, показателем оценивания – глубина и качество отработанных вопросов, оформление работы.

**«Отлично»** – работа написана грамотным и правильным языком. Цели поставлены ясно, всесторонне раскрыты и полностью соответствуют теме. Работа имеет логическую связанность и цельность, отражено собственное мнение обучающегося, хорошо обоснованы выводы.

**«Хорошо»** – текст написан грамотным языком. Цели и задачи вполне раскрыты и в основном соответствуют теме. Хорошо подобрана аргументация, правильно используются факты. Применяется объяснительный стиль с элементами критической интерпретации, не до конца сформировано собственное мнение обучающегося.

**«Удовлетворительно»** – оформление соответствует требованиям руководящих документов, но тема не раскрыта. Имеются поверхностные суждения, слабо отражено собственное мнение обучающегося.

**«Неудовлетворительно»** – работа имеет отдельные части, которые в логическое целое не связаны. Цели ограничены. Отмечается недостаточное понимание фактов и проблем. Тема не раскрыта. Не сформировано собственное мнение обучающегося.



## **Пример подготовки к эссе**

Рассмотрим порядок написания эссе и его возможное содержание на примере темы «Классификация языков программирования». В первую очередь выделяем предметные понятия дисциплины «Программирование» в данной теме, это языки программирования. Вспоминаем определение этого термина, собираем в литературных и сетевых источниках его различные трактовки. Находим в них общее, пытаемся понять, на основе каких аспектов сформулировано то или иное определение, какие характерные черты взяты за основу. На основе собранной информации формулируем собственное определение, отражающее наш взгляд на это понятие. Таким образом прорабатывается каждое определение из темы эссе.

Далее анализируем нашу тему. В ней говорится о классификации языков программирования. Подбираем в литературных и сетевых источниках различные точки зрения на роль информационных технологий, анализируем информацию и пытаемся сформулировать собственное мнение по этому вопросу. Особое внимание стоит уделить точкам зрения, которые противоречат друг другу, в этом противоречии потенциально можно найти проблему. Формулируем собственное мнение по данному вопросу.

Таким образом, мы рассмотрели в источниках с разных точек зрения все аспекты, входящие в нашу тему эссе. Далее, основываясь на сделанном анализе, мы формулируем проблему, которая кажется нам наиболее актуальной, и отражаем в эссе нашу точку зрения на нее. При этом может быть полезным следующий прием: после того, как мы сформировали свое мнение, возвращаемся к анализу источников и изучаем те из них, которые высказывают отличную от нашей точку зрения. Пытаемся понять ее, возможно, нами было что-то упущено, или же наоборот, мы лишь сильнее утвердимся в своем решении.

## **РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ К ТЕСТИРОВАНИЮ**

Тесты являются эффективным средством контроля образовательных результатов на уровне знания и понимания. Тестирование по дисциплине «**Программирование**» позволяет оценить уровень освоения обучающимся основных теоретических понятий, изучаемых в курсе, и взаимосвязей между ними. Критерием оценки теста являются правильные ответы на поставленные вопросы, показателем оценивания – процент верных ответов на вопросы, шкала оценивания (оценка) – выделено 4 уровня:

высокий (отлично) - более 80% правильных ответов;

достаточный (хорошо)– от 60 до 80 % правильных ответов;

пороговый(удовлетворительно)– от 50 до 60% правильных ответов;

критический(неудовлетворительно)– менее 50% правильных ответов.

Для подготовки к тестированию рекомендуется повторить предметные понятия из конспектов лекций, как в прямом порядке, т.е. по названию термина даем его определение, так и в обратном, т.е. по определению называем, о каком термине идет речь. Далее выявляем связи (общее-частное, предшествующее-последующее и т.д.) между этими понятиями.

## **РЕКОМЕНДАЦИИ ПО ПОДГОТОВКЕ И НАПИСАНИЮ КОНТРОЛЬНЫХ РАБОТ**

Текущий контроль по дисциплине предусматривает написание двух контрольных работ, направленных на проверку уровня усвоения материала тем 2 и 3. Контрольная работа – это самостоятельная аналитическая письменная работа, которая способствует углубленному изучению пройденного теоретического материала. Цель контрольной работы – получить практические знания, умения и навыки по написанию программ на языке высокого уровня.

Задание в контрольной работе построено по схеме «А-знает, Б-умеет, В-владеет», т.е. для ответа на вопрос А достаточно знать базовый теоретический материал по теме задания, для ответа на вопрос части Б

требуется умение создавать из базовых теоретических сведений собственные абстракции, а часть В проверяет способность применения полученных знаний и умений к решению практической задачи.

Критерием оценки контрольной работы является правильность решения задач, показателем оценивания – процент правильно решенных задач. Каждая часть задания оценивается по своей шкале:

Часть А – 1 балл, если дан верный ответ; 0.5 балла, если ответ в целом верный, но есть небольшие неточности; 0 баллов иначе.

Часть Б – 2 балла, если дан верный ответ; 1 балл, если ответ в целом верный, но есть небольшие неточности; 0.5 балла, если ответ в основном неверный, 0 баллов иначе.

Часть В – 2 балла, если задача решена верно; 1 балл, если задача решена в целом верно, но есть небольшие неточности; 0 баллов иначе.

Итоговый балл за задание находится суммированием баллов за все части:

$$\text{ИБ} = \text{А} + \text{Б} + \text{В}.$$

В контрольной работе по теме 2 всего 5 заданий, таким образом, за все задания максимально можно получить 25 баллов. В оценку по пятибалльной шкале эти баллы переводятся по схеме:

менее 10 баллов – «неудовлетворительно»;

от 10 до 14 баллов – «удовлетворительно»;

от 14 баллов до 20 – «хорошо»;

более 20 баллов – «отлично».

В контрольной работе по теме 3 одно задание, которое оценивается по приведенной выше шкале.

### **Вопросы для подготовки к контрольной работе**

#### **Тема 2. Основы языка Delphi**

- 1) Типы данных в Delphi
- 2) Условный оператор в Delphi
- 3) Операторы цикла в Delphi

- 4) Подпрограммы в Delphi
- 5) Массивы в Delphi
- 6) Сортировка массивов
- 7) Файлы в Delphi
- 8) Множества в Delphi

### **Тема 3. Основы объектно-ориентированного программирования**

- 1) Классы и их составляющие в Delphi
- 2) Поля, методы и свойства
- 3) Инкапсуляция
- 4) Полиморфизм
- 5) Работа с объектами в Delphi

## **РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА**

### **основная**

1. Ачкасов, В. Ю. Введение в программирование на Delphi / В. Ю. Ачкасов. — 2-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 295 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/73666.html>.
2. Ачкасов, В. Ю. Программирование на Lazarus / В. Ю. Ачкасов. — 2-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 520 с. — ISBN 2227-8397. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <http://www.iprbookshop.ru/73711.html>.

### **дополнительная**

3. Архитектура информационных систем: учебное пособие для вузов. Рыбальченко М.В.. М.: Издательство Юрайт, 2018. Режим доступа: ЭБС Юрайт, URL: <https://www.biblio-online.ru/bcode/415069>.

4. Трофимов, В. В. Алгоритмизация и программирование : учебник для академического бакалавриата / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2018. — 137 с. — (Бакалавр. Академический курс. Модуль). — ISBN 978-5-9916-9866-5. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://www.biblio-online.ru/bcode/414652>.