


МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
БОРИСОГЛЕБСКИЙ ФИЛИАЛ
(БФ ФГБОУ ВО «ВГУ»)

УТВЕРЖДАЮ

Заведующий кафедрой
естественнонаучных и
общеобразовательных дисциплин

 С.Е. Зюзин

01.09.2018 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Б1.В.14 Программирование

1. Шифр и наименование направления подготовки:

44.03.05 Педагогическое образование (с двумя профилями подготовки)

2. Профили подготовки:

Математика. Информатика и информационные технологии в образовании

3. Квалификация выпускника:

Бакалавр

4. Форма обучения:

Очная, заочная

5. Кафедра, отвечающая за реализацию дисциплины:

естественнонаучных и общеобразовательных дисциплин

6. Составитель программы:

В. В. Волков, кандидат физико-математических наук, доцент

7. Рекомендована:

научно-методическим советом Филиала (протокол № 1 от 31.08.2018 г.)

8. Семестры: 5,6 (ОФО); 7, 8 (ЗФО)

9. Цель и задачи учебной дисциплины:

Целью учебной дисциплины «Программирование» является формирование систематизированных знаний и навыков в области программирования: ознакомление студентов с историей развития языков программирования, с парадигмами программирования, структурами данных, знакомство с методами, применяемыми в программировании, известными алгоритмами, возможностями работы с динамической памятью.

Задачи учебной дисциплины:

- изучение основных конструкций языка программирования процедурного типа на примере Pascal;
- изучение алгоритмов обработки массивов;
- обучение построению и использованию модулей;
- знакомство с динамическим распределением памяти и указателями;
- обучение работе со списками, деревьями, очередями;
- объектно-ориентированный подход на примере языка программирования Free Pascal.

10. Место учебной дисциплины в структуре образовательной программы:

Дисциплина «Программирование» входит в блок Б1 «Дисциплины (модули)» и является обязательной дисциплиной вариативной части образовательной программы.

Для освоения дисциплины «Программирование» студенты используют знания, умения, навыки, сформированные в ходе изучения дисциплин «Информатика», «Основы программирования».

Изучение данной дисциплины является необходимой основой для последующего изучения дисциплин, «Теоретические основы информатики», «Архитектура компьютера», «Информационные системы», «Основы искусственного интеллекта», «Методика обучения информатике», «Компьютерные сети, Интернет и мультимедиа технологии».

Понятийный, методологический и технологический материал курса играет важную роль в формировании научного мировоззрения будущего учителя, его информационной грамотности.

Условия реализации дисциплины для лиц с ОВЗ определяются особенностями восприятия учебной информации и с учетом индивидуальных психофизических особенностей.

11. Планируемые результаты обучения по дисциплине/модулю (знания, умения, навыки), соотнесенные с планируемыми результатами освоения образовательной программы (компетенциями выпускников):

Компетенция		Планируемые результаты обучения
Код	Название	
ПК-1	готовность реализовывать образовательные программы по учебным предметам в соответствии с требованиями образовательных стандартов	знает (имеет представление): – связь теоретических основ и технологических приёмов учебной дисциплины с содержанием преподаваемых учебных предметов (<i>основы программирования на языке Pascal; приемы создания приложений в среде PascalABC.NET (Free Pascal, Geany); основы объектно-ориентированного программирования; приемы создания приложений визуальными средствами в ИСР Lazarus; способы определения стандартных и пользовательских типов данных; основы модульного проектирования программ; методы динамического размещения в памяти переменных и массивов стандартных типов, а также экземпляров структур, массивов структур, объектов классов и массивов объектов классов; способы объявления указателей и алгоритмы работы с ними; динамические линейные структуры</i>

		<p>данных (списки, очереди, стеки), способы их представления в памяти и обработки в программе; динамические нелинейные структуры данных (графы, деревья), способы их представления в памяти и обработки в программе; методы и средства обработки исключительных ситуаций);</p> <p>умеет:</p> <ul style="list-style-type: none"> – применять навыки владения ИКТ, проектной и исследовательской деятельностью в процессе изучения учебной дисциплины; – осуществлять деятельность по разработанным программам учебных предметов; <p>имеет навыки:</p> <ul style="list-style-type: none"> – владения профессиональным инструментарием, позволяющим реализовывать учебные программы в соответствии с требованиями образовательных стандартов;
ПК-4	<p>способность использовать возможности образовательной среды для достижения личностных, метапредметных и предметных результатов обучения и обеспечения качества учебно-воспитательного процесса средствами преподаваемых учебных предметов</p>	<p>знает:</p> <ul style="list-style-type: none"> – технологические приемы преподаваемого учебного предмета, лежащие в основе построения различных моделей в экономике, социологии, эконометрике и т.д. (<i>основы алгоритмизации, простые алгоритмы поиска и сортировки данных</i>); <p>умеет:</p> <ul style="list-style-type: none"> – использовать знание основ учебной дисциплины для перевода информации с естественного языка на язык соответствующей предметной области и обратно (<i>создавать программы средствами языка программирования Pascal; создавать приложения в визуальной среде Lazarus</i>); – применять теоретические знания по учебной дисциплине в описании процессов и явлений в различных областях знания (<i>применять технологию структурного программирования для решения задач; применять технологию объектно-ориентированного программирования для решения задач</i>); – использовать преимущества технологических приемов учебной дисциплины при решении задач преподаваемых учебных предметов; – осуществлять поиск и отбор информации, необходимой для решения конкретной задачи; <p>владеет:</p> <ul style="list-style-type: none"> – материалом учебной дисциплины на уровне, позволяющем формулировать и решать задачи, возникающие в ходе учебной деятельности по преподаваемым предметам, а также в практической деятельности, требующие углубленных профессиональных знаний (<i>понятийным аппаратом структурно-алгоритмического (модульного), объектно-ориентированного и визуального программирования; визуальной и объектно-ориентированной парадигмой программирования; современными технологиями разработки надежного программного кода</i>); – навыками формализации теоретических и прикладных практических задач.

12. Объем дисциплины в зачетных единицах/часах — 9 / 324.

Форма промежуточной аттестации: зачет с оценкой, экзамен.

13. Виды учебной работы

Очная форма обучения

Вид учебной работы	Трудоемкость		
	Всего	По семестрам	
		5	6
Контактная работа, в том числе:	152	72	80
лекции	38	18	20
практические занятия	38	18	20
лабораторные работы	76	36	40
Самостоятельная работа	136	36	100
Форма промежуточной аттестации (зачет с оценкой – 0 час., экзамен – 36 час.)	36	0	36
Итого:	324	108	216

Заочная форма обучения

Вид учебной работы	Трудоемкость		
	Всего	По семестрам	
		7	8
Контактная работа, в том числе:	32	18	14
лекции	8	4	4
практические занятия	4	4	0
лабораторные работы	20	10	10
Самостоятельная работа	279	86	193
Форма промежуточной аттестации (зачет с оценкой – 4 час., экзамен – 9 час.)	13	4	9
Итого:	324	108	216

13.1. Содержание дисциплины

Очная форма обучения

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
1. Лекции		
1.1	Структурированные типы данных языка программирования Pascal.	Структурированный тип – массив. Сортировка массивов. Двумерные массивы. Строки. Обработка строк. Тип-файл.
1.2	Процедурное программирование.	Подпрограммы: основные понятия, фактические и формальные параметры. Функции. Формат записи. Процедуры. Формат записи. Рекурсия.
1.3	Модульное программирование. Графика.	Модульное программирование. Структура модуля. Модули, подключение модулей к программе. Использование библиотек подпрограмм. Графика в Pascal. Графические операторы.
1.4	Динамические структуры данных.	Понятие динамических структур данных. Указатели. Объявление. Подпрограммы для работы с указателями. Линейный односвязный список: просмотр, добавление, удаление элементов. Стек. Очередь. Бинарные деревья.
1.5	Объектно-ориентированное программирование.	Объектно-ориентированная парадигма программирования. Объекты, полиморфизм и наследование. Объектно-ориентированное проектирование. Визуальные среды программирования. Конструирование

		программ на основе иерархии объектов. Сообщения. Обработка исключительных ситуаций.
2. Практические занятия		
2.1	Структурированные типы данных языка программирования Pascal.	Структурированный тип – массив. Сортировка массивов. Двумерные массивы. Строки. Обработка строк. Тип-файл.
2.2	Процедурное программирование.	Подпрограммы: основные понятия, фактические и формальные параметры. Функции. Формат записи. Процедуры. Формат записи. Рекурсия.
2.3	Динамические структуры данных.	Указатели. Объявление. Подпрограммы для работы с указателями. Линейный односвязный список: просмотр, добавление, удаление элементов. Стек. Очередь. Бинарные деревья. Использование динамических структур данных при решении задач: проверка правильности расстановки скобок, перевод выражения в постфиксную форму.
2.4	Объектно-ориентированное программирование.	Объектно-ориентированная парадигма программирования. Объекты, полиморфизм и наследование. Визуальные среды программирования. Конструирование программ на основе иерархии объектов. Обработка исключительных ситуаций.
3. Лабораторные работы		
3.1	Структурированные типы данных языка программирования Pascal.	Структурированный тип – массив. Сортировка массивов. Двумерные массивы. Строки. Обработка строк. Тип-файл.
3.2	Процедурное программирование.	Функции. Процедуры. Рекурсия. Использование подпрограмм в решении математических задач.
3.3	Модульное программирование. Графика.	Модульное программирование. Модули, подключение модулей к программе. Графика в Pascal. Графические операторы. Простейшая анимация.
3.4	Динамические структуры данных.	Указатели. Объявление. Подпрограммы для работы с указателями. Линейный односвязный список: просмотр, добавление, удаление элементов. Стек. Очередь. Бинарные деревья. Использование динамических структур данных при решении задач: проверка правильности расстановки скобок, перевод выражения в постфиксную форму.
3.5	Объектно-ориентированное программирование.	Объектно-ориентированное проектирование. Визуальные среды программирования. Конструирование программ на основе иерархии объектов. Обработка исключительных ситуаций.

Заочная форма обучения

№ п/п	Наименование раздела дисциплины	Содержание раздела дисциплины
1. Лекции		
1.1	Структурированные типы данных языка программирования Pascal.	Структурированный тип – массив. Сортировка массивов. Двумерные массивы. Строки. Обработка строк. Тип-файл.
1.2	Процедурное программирование.	Подпрограммы: основные понятия, фактические и формальные параметры. Функции. Формат записи. Процедуры. Формат записи. Рекурсия.
1.3	Модульное программирование. Графика.	Модульное программирование. Структура модуля. Модули, подключение модулей к программе. Использование библиотек подпрограмм. Графика в Pascal. Графические операторы. Простейшая

		анимация.
1.4	Динамические структуры данных.	Понятие динамических структур данных. Указатели. Объявление. Подпрограммы для работы с указателями. Линейный односвязный список: просмотр, добавление, удаление элементов. Стек. Очередь. Бинарные деревья.
1.5	Объектно-ориентированное программирование.	Объектно-ориентированная парадигма программирования. Объекты, полиморфизм и наследование. Объектно-ориентированное проектирование. Визуальные среды программирования. Конструирование программ на основе иерархии объектов. Сообщения. Обработка исключительных ситуаций.
2. Практические занятия		
2.1	Структурированные типы данных языка программирования Pascal.	Структурированный тип – массив. Сортировка массивов. Двумерные массивы. Строки. Обработка строк. Тип-файл.
2.2	Процедурное программирование.	Подпрограммы: основные понятия, фактические и формальные параметры. Функции. Формат записи. Процедуры. Формат записи. Рекурсия.
3. Лабораторные работы		
3.1	Структурированные типы данных языка программирования Pascal.	Структурированный тип – массив. Сортировка массивов. Двумерные массивы. Строки. Обработка строк. Тип-файл.
3.2	Процедурное программирование.	Функции. Процедуры. Рекурсия. Использование подпрограмм в решении математических задач.
3.3	Модульное программирование. Графика.	Модульное программирование. Модули, подключение модулей к программе. Графика в Pascal. Графические операторы. Простейшая анимация.
3.4	Динамические структуры данных.	Указатели. Объявление. Подпрограммы для работы с указателями. Линейный односвязный список: просмотр, добавление, удаление элементов. Стек. Очередь. Бинарные деревья. Использование динамических структур данных при решении задач: проверка правильности расстановки скобок, перевод выражения в постфиксную форму.
3.5	Объектно-ориентированное программирование.	Объектно-ориентированное проектирование. Визуальные среды программирования. Конструирование программ на основе иерархии объектов. Обработка исключительных ситуаций.

13.2. Темы (разделы) дисциплины и виды занятий

Очная форма обучения

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				
		Лекции	Практические занятия	Лабораторные работы	Самостоятельная работа	Всего
5 семестр						
1	Структурированные типы данных языка программирования Pascal.	6	8	12	12	38
2	Процедурное программирование.	6	10	12	12	40
3	Модульное программирование. Графика.	6	0	12	12	30
	Зачёт с оценкой					0
	Всего в 5 семестре:	18	18	36	36	108
6 семестр						

4	Динамические структуры данных.	10	10	20	50	90
5	Объектно-ориентированное программирование.	10	10	20	50	90
	Экзамен					36
	Всего в 6 семестре:	20	20	40	100	216
	Итого:	38	38	76	136	324

Заочная форма обучения

№ п/п	Наименование темы (раздела) дисциплины	Виды занятий (часов)				Всего
		Лекции	Практические занятия	Лабораторные работы	Самостоятельная работа	
7 семестр						
1	Структурированные типы данных языка Pascal.	1	2	2	26	31
2	Процедурное программирование.	1	2	4	30	37
3	Модульное программирование. Графика.	2	0	4	30	36
	Зачёт с оценкой					4
	Всего в 7 семестре:	4	4	10	86	108
8 семестр						
4	Динамические структуры данных.	2	0	4	100	106
5	Объектно-ориентированное программирование.	2	0	6	93	101
	Экзамен					9
	Всего в 8 семестре:	4	0	10	193	216
	Итого:	8	4	20	279	324

14. Методические указания для обучающихся по освоению дисциплины

Приступая к изучению учебной дисциплины, прежде всего обучающиеся должны ознакомиться с учебной программой дисциплины. Электронный вариант рабочей программы размещён на сайте БФ ВГУ.

Знание основных положений, отраженных в рабочей программе дисциплины, поможет обучающимся ориентироваться в изучаемом курсе, осознавать место и роль изучаемой дисциплины в подготовке будущего педагога, строить свою работу в соответствии с требованиями, заложенными в программе.

Основными формами контактной работы по дисциплине являются лекции, практические и лабораторные занятия, посещение которых обязательно для всех студентов (кроме студентов, обучающихся по индивидуальному плану).

В ходе лекционных занятий следует не только слушать излагаемый материал и кратко его конспектировать, но очень важно участвовать в анализе примеров, предлагаемых преподавателем, в рассмотрении и решении проблемных вопросов, выносимых на обсуждение. Необходимо критически осмысливать предлагаемый материал, задавать вопросы как уточняющего характера, помогающие уяснить отдельные излагаемые положения, так и вопросы продуктивного типа, направленные на расширение и углубление сведений по изучаемой теме, на выявление недостаточно освещенных вопросов, слабых мест в аргументации и т.п.

Не следует дословно записывать лекцию, лучше попытаться понять логику изложения и выделить наиболее важные положения лекции в виде опорного конспекта. Рекомендуется использовать различные формы выделения наиболее сложно-

го, нового, непонятного материала, который требует дополнительной проработки: можно пометить его знаком вопроса (или записать на полях сам вопрос), цветом, размером букв и т.п. – это поможет быстро найти материал, вызвавший трудности, и в конце лекции (или сразу же, попутно) задать вопрос преподавателю (не следует оставлять непонятый материал без дополнительной проработки, без него иногда бывает невозможно понять последующие темы). Материал уже знакомый или понятный нуждается в меньшей детализации – это поможет сэкономить усилия во время конспектирования.

На практических занятиях рекомендуется активно участвовать в анализе решаемых задач, обсуждении алгоритма их решения, выборе способов реализации алгоритма на языке программирования. При возникновении затруднений в решении задач важно сразу выяснить все непонятные моменты, задав вопрос преподавателю.

В ходе выполнения лабораторных работ рекомендуется пользоваться конспектами лекций и записями с практических занятий. При необходимости, за справочной информацией по языку программирования рекомендуется обращаться к встроенной справке среды разработки или к онлайн-справочникам. Важно при решении задач придерживаться правил стилового оформления кода: это сделает код более «читаемым», поможет в его анализе (и поиске ошибок при необходимости).

При подготовке к промежуточной аттестации необходимо повторить пройденный материал в соответствии с учебной программой, примерным перечнем вопросов, выносящихся на аттестацию. Рекомендуется использовать конспекты лекций и источники, перечисленные в списке литературы в рабочей программе дисциплины, а также ресурсы электронно-библиотечных систем. Необходимо обратить особое внимание на темы учебных занятий, пропущенных по разным причинам. При необходимости можно обратиться за консультацией и методической помощью к преподавателю.

Для достижения планируемых результатов обучения используются интерактивные лекции, групповые дискуссии.

15. Перечень основной и дополнительной литературы, ресурсов интернет, необходимых для освоения дисциплины

а) основная литература:

№ п/п	Источник
1	Долинер, Л. И. Основы программирования в среде PascalABC.NET: учебное пособие [Электронный ресурс] / Л. И. Долинер. — Екатеринбург: Изд-во Уральского университета, 2014. — 129 с. — URL: http://biblioclub.ru/index.php?page=book&id=275988 (29.06.2018)
2	Павлова, Т. Ю. Структурное программирование в ИСР «Free Pascal»: учебное пособие [Электронный ресурс] / Т. Ю. Павлова. — Кемерово: КГУ, 2010. — 88 с. — URL: http://biblioclub.ru/index.php?page=book&id=232450 (29.06.2018)
3	Немцова, Т. И. Программирование на языке Object Pascal: учебное пособие [Текст] / Т. И. Немцова, С. Ю. Голова, И. В. Абрамова. — М.: ИНФРА-М, 2009. — 496 с.

б) дополнительная литература:

№ п/п	Источник
4	Александрова, Л. В. Основы программирования на языке Паскаль: учебно-методическое пособие [Электронный ресурс] / Л. В. Александрова, А. М. Мардашев, Е. Н. Матюхина. — М.: Российский университет дружбы народов, 2013. — 116 с. — URL: http://biblioclub.ru/index.php?page=book&id=226863 (29.06.2018)
5	Беляева, И. В. Основы программирования на языке Turbo Pascal: учебное пособие [Электронный ресурс] / И. В. Беляева. — Ульяновск: УлГТУ, 2011. — 266 с. — URL: http://window.edu.ru/resource/529/74529 (29.06.2018)

в) информационные электронно-образовательные ресурсы (официальные ресурсы интернет):

№ п/п	Источник
-------	----------

6	Грацианова, Т. Ю. Программирование в примерах и задачах [Электронный ресурс] / Т. Ю. Грацианова. — 2-е изд. — М.: БИНОМ. Лаборатория знаний, 2015. — 354 с. — URL: http://biblioclub.ru/index.php?page=book&id=362860 (29.06.2018)
7	Ачкасов, В. Программирование на Lazarus [Электронный ресурс] / В. Ачкасов. — 2-е изд., исправ. — М.: Национальный Открытый Университет «ИНТУИТ», 2016. — 521 с. — URL: http://biblioclub.ru/index.php?page=book&id=429187 (29.06.2018)
8	Алексеев, Е. Программирование на Free Pascal и Lazarus: курс [Электронный ресурс] / Е. Алексеев, О. Чеснокова, Т. Кучер. — 2-е изд., исправ. — М.: Национальный Открытый Университет «ИНТУИТ», 2016. — 552 с. — URL: http://biblioclub.ru/index.php?page=book&id=429189 (29.06.2018)
9	ЭУК «Программирование (6 семестр)», https://edu.vsu.ru/course/view.php?id=6159

16. Перечень учебно-методического обеспечения для самостоятельной работы

№ п/п	Источник
1	Грацианова, Т. Ю. Программирование в примерах и задачах [Электронный ресурс] / Т. Ю. Грацианова. — 2-е изд. — М.: БИНОМ. Лаборатория знаний, 2015. — 354 с. — URL: http://biblioclub.ru/index.php?page=book&id=362860 (29.06.2018);
2	Алексеев, Е. Программирование на Free Pascal и Lazarus: курс [Электронный ресурс] / Е. Алексеев, О. Чеснокова, Т. Кучер. — 2-е изд., исправ. — М.: Национальный Открытый Университет «ИНТУИТ», 2016. — 552 с. — URL: http://biblioclub.ru/index.php?page=book&id=429189 (29.06.2018);
3	План лекционных занятий (размещён на сайте филиала)
4	Методические указания к выполнению лабораторных работ (ресурсный фонд кафедры)

17. Информационные технологии, используемые для реализации учебной дисциплины, включая программное обеспечение, информационно-справочные системы и профессиональные базы данных

Программное обеспечение:

Операционные системы и их оболочки: Microsoft Windows.

При реализации дисциплины применяется смешанное обучение с использованием:

- ЭУК «Программирование (6 семестр)», <https://edu.vsu.ru/course/view.php?id=6159>;
- онлайн-консультаций;
- мессенджеров (<https://vk.com>);
- электронной почты.

Информационно-справочные системы и профессиональные базы данных:

–Федеральный портал Российское образование – <http://www.edu.ru/>

–Информационная система «Единое окно доступа к образовательным ресурсам» <http://window.edu.ru/>

–Лекции ведущих преподавателей вузов России в свободном доступе – <https://www.lektorium.tv/>

–Электронно-библиотечная система «Университетская библиотека online» – <http://biblioclub.ru/>

18. Материально-техническое обеспечение дисциплины:

Мультимедийное оборудование (проектор, ноутбук или стационарный компьютер, экран), компьютерный класс (компьютеры, объединенные в сеть с выходом в Интернет и обеспечением доступа в электронную информационно-образовательную среду ВГУ и БФ).

19. Фонд оценочных средств:

19.1 Перечень компетенций с указанием этапов формирования и планируемых результатов обучения

Код и содержание компетенции (или ее части)	Планируемые результаты обучения (показатели достижения заданного уровня освоения компетенции посредством формирования знаний, умений, навыков)	Этапы формирования компетенции (разделы (темы) дисциплины или модуля и их наименование)	Оценочные материалы для проведения текущего контроля успеваемости, промежуточной аттестации обучающихся
ПК-1 готовность реализовать образовательные программы по учебным предметам в соответствии с требованиями образовательных стандартов	<p>Знать:</p> <p>– связь теоретических основ и технологических приёмов учебной дисциплины с содержанием преподаваемых учебных предметов (<i>основы программирования на языке Pascal; основы объектно-ориентированного программирования; приемы создания приложений визуальными средствами в ИСР Lazarus; способы определения стандартных и пользовательских типов данных; основы модульного проектирования программ; методы динамического размещения в памяти переменных и массивов стандартных типов, а также экземпляров структур, массивов структур, объектов классов и массивов объектов классов; способы объявления указателей и алгоритмы работы с ними; динамические линейные структуры данных (списки, очереди, стеки), способы их представления в памяти и обработки в программе; динамические нелинейные структуры данных (графы, деревья), способы их представления в памяти и обработки в программе; методы и средства обработки исключительных ситуаций</i>);</p>	<p>Структурированные типы данных языка программирования Pascal. Процедурное программирование. Модульное программирование. Графика. Динамические структуры данных. Объектно-ориентированное программирование.</p>	<p>Задачи к Л/р №1, Задачи к Л/р №2, Задачи к Л/р №3, Контрольные вопросы к Модулю №1 Лабораторного практикума, Контрольные вопросы к Модулю №2 Лабораторного практикума, Контрольные вопросы к Модулю №3 Лабораторного практикума, Контрольные вопросы к Лабораторной работе «Основы Lazarus»</p>
	<p>Уметь:</p> <p>- применять навыки владения ИКТ, проектной и исследовательской деятельностью в процессе изучения учебной дисциплины; – осуществлять деятельность по разработанным программам учебных предметов;</p>	<p>Структурированные типы данных языка программирования Pascal. Процедурное программирование. Модульное программирование. Графика. Динамические структуры данных. Объектно-ориентированное программирование.</p>	<p>Задачи к Л/р №1, Задачи к Л/р №2, Задачи к Л/р №3, Контрольные вопросы к Модулю №1 Лабораторного практикума, Контрольные вопросы к Модулю №2 Лабораторного практикума, Контрольные вопросы к Модулю №3 Лабораторного практикума, Контрольные вопросы к Лабораторной работе «Основы Lazarus», Темы индивидуальных проектов</p>

			<p>Задачи к Л/р №1, Задачи к Л/р №2, Задачи к Л/р №3, Контрольные вопросы к Модулю №1 Лабораторного практикума, Контрольные вопросы к Модулю №2 Лабораторного практикума, Контрольные вопросы к Модулю №3 Лабораторного практикума, Контрольные вопросы к Лабораторной работе «Основы Lazarus», Темы индивидуальных проектов</p>
	<p>Владеть: – профессиональным инструментарием, позволяющим реализовывать учебные программы в соответствии с требованиями образовательных стандартов;</p>	<p>Структурированные типы данных языка программирования Pascal. Процедурное программирование. Модульное программирование. Графика. Динамические структуры данных. Объектно-ориентированное программирование.</p>	
<p>ПК-4 способность использовать возможности образовательной среды для достижения личностных, метапредметных и предметных результатов обучения и обеспечения качества учебно-воспитательного процесса средствами преподаваемых учебных предметов</p>	<p>Знать: – технологические приемы преподаваемого учебного предмета, лежащие в основе построения различных моделей в экономике, социологии, эконометрике и т.д. (<i>основы алгоритмизации, простые алгоритмы поиска и сортировки данных</i>);</p>	<p>Структурированные типы данных языка программирования Pascal. Процедурное программирование. Модульное программирование. Графика. Динамические структуры данных. Объектно-ориентированное программирование.</p>	<p>Задачи к Л/р №1, Задачи к Л/р №2, Задачи к Л/р №3, Контрольные вопросы к Лабораторной работе «Основы Lazarus»</p>
	<p>Уметь: – использовать знание основ учебной дисциплины для перевода информации с естественного языка на язык соответствующей предметной области и обратно (<i>создавать программы средствами языка программирования Pascal; создавать приложения в визуальной среде Lazarus</i>); – применять теоретические знания по учебной дисциплине в описании процессов и явлений в различных областях знания (<i>применять технологию структурного программирования для решения задач; применять технологию объектно-ориентированного программирования для решения задач</i>); – использовать преимущества технологических приемов учебной дисциплины при решении задач преподаваемых учебных предметов; осуществлять поиск и отбор информации, необходимой для решения конкретной задачи;</p>	<p>Структурированные типы данных языка программирования Pascal. Процедурное программирование. Модульное программирование. Графика. Динамические структуры данных. Объектно-ориентированное программирование.</p>	<p>Задачи к Л/р №1, Задачи к Л/р №2, Задачи к Л/р №3, Контрольные вопросы к Модулю №1 Лабораторного практикума, Контрольные вопросы к Модулю №2 Лабораторного практикума, Контрольные вопросы к Модулю №3 Лабораторного практикума, Контрольные вопросы к Лабораторной работе «Основы Lazarus», Темы индивидуальных проектов</p>

	<p>Владеть:</p> <ul style="list-style-type: none"> – материалом учебной дисциплины на уровне, позволяющем формулировать и решать задачи, возникающие в ходе учебной деятельности по преподаваемым предметам, а также в практической деятельности, требующие углубленных профессиональных знаний (<i>понятийным аппаратом структурно-алгоритмического (модульного), объектно-ориентированного и визуального программирования; визуальной и объектно-ориентированной парадигмой программирования; современными технологиями разработки надежного программного кода</i>); – навыками формализации теоретических и прикладных практических задач. 	<p>Структурированные типы данных языка программирования Pascal. Процедурное программирование. Модульное программирование. Графика. Динамические структуры данных. Объектно-ориентированное программирование.</p>	<p>Задачи к Л/р №1, Задачи к Л/р №2, Задачи к Л/р №3, Контрольные вопросы к Модулю №1 Лабораторного практикума, Контрольные вопросы к Модулю №2 Лабораторного практикума, Контрольные вопросы к Модулю №3 Лабораторного практикума, Контрольные вопросы к Лабораторной работе «Основы Lazarus», Темы индивидуальных проектов</p>
Промежуточная аттестация 1 – зачет с оценкой			По результатам работы в семестре
Промежуточная аттестация 2 – экзамен			Вопросы к экзамену

19.2 Описание критериев и шкалы оценивания компетенций (результатов обучения) при промежуточной аттестации

Для оценивания результатов обучения на зачете с оценкой и экзамене используется 4-балльная шкала: «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Соотношение показателей, критериев и шкалы оценивания результатов обучения.

Критерии оценивания компетенций	Уровень сформированности компетенций	Шкала оценок
<i>Обучающийся в полной мере владеет теоретическими основами дисциплины, обнаруживает свободное владение понятийным аппаратом, способен иллюстрировать ответ примерами, фактами, применять теоретические знания для решения практических задач в области программирования, демонстрирует полное освоение показателей формируемых компетенций</i>	<i>Повышенный уровень</i>	<i>Отлично</i>
<i>Обучающийся владеет теоретическими основами дисциплины, имеет представление об основных подходах к излагаемому материалу, знает определения основных теоретических понятий излагаемой темы, в основном демонстрирует готовность применять теоретические знания в практической деятельности и освоение большинства показателей формируемых компетенций</i>	<i>Базовый уровень</i>	<i>Хорошо</i>
<i>Обучающийся владеет частично теоретическими основами дисциплины, в целом имеет представление об основных понятиях излагаемой темы, фрагментарно способен применять теоретические знания в практической деятельности, демонстрирует освоение некоторых показателей формируемых компетенций</i>	<i>Пороговый уровень</i>	<i>Удовлетворительно</i>
<i>Обучающийся демонстрирует отрывочные, фрагментарные знания, не ориентируется в теоретическом материале, допускает грубые ошибки при ответе, не демонстрирует готовность применять теоретические знания в практической деятельности и освоение показателей формируемых компетенций.</i>	–	<i>Неудовлетворительно</i>

19.3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы .

19.3.1 Задачи к Лабораторной работе №1

1. Объявите 3 массива: A, B, C. Массив A – таблица целых чисел размером 2x3. Массив B – таблица целых чисел размером 5x5. Массив C – таблица действительных чисел размером 10x4. Массивы A и B должны объявляться без использования нового типа данных, а массив C должен быть объявлен как вновь созданный тип данных.
2. Заполните объявленные массивы. Массив A заполните, используя клавиатуру. Примените приглашение к вводу.
3. Массив B заполните, используя генератор случайных чисел, так чтобы массив содержал числа от -99 до 99. Массив C заполните по формуле: $C_{i,j} = \sin(5i) - \cos(3j)$.
4. Выведите полученные массивы на экран. Массив A выведите на экран в виде столбца следующего формата:
5. $A[\langle \text{индекс 1} \rangle, \langle \text{индекс 2} \rangle] = \langle \text{значение:5} \rangle$.
6. Например: $A[1,2] = 4$. Массивы B и C выведите в виде таблицы. Предусмотрите возможность задержки работы программы после вывода каждого из массивов.
7. Подсчитайте сумму элементов массива A и выведите ее на экран.
8. Для элементов массива B найдите среднее арифметическое. Результат выведите на экран.
9. Для положительных элементов массива B найдите среднее арифметическое. Результат выведите на экран.
10. Найдите сумму квадратов элементов массива B кратных 4. Результат выведите на экран.
11. Найдите минимальный элемент массива A. Номер и значение данного элемента выведите на экран.
12. В массиве C поменяйте местами следующие элементы: $C_{1,1}$ и $C_{5,2}$, $C_{10,4}$ и $C_{7,3}$. Полученный массив выведите на экран. Предусмотрите возможность задержки работы программы.
13. Найдите сумму элементов массива C, находящихся на главной диагонали. Результат выведите на экран.

Критерии оценки:

- оценка «зачтено» выставляется студенту, если задание выполнено полностью или с незначительными недочётами;
- оценка «не зачтено» выставляется студенту, если задание не выполнено или выполнено с серьёзными недочётами.

19.3.2 Задачи к Лабораторной работе №2

1. Напишите программу вычисления

$$\left\{ \sum_{k=1}^n \frac{(-1)^k}{(3k^2)!!}, \text{ где } n!! = \begin{cases} 1 \cdot 3 \cdot 5 \cdot \dots \cdot n, & \text{если } n - \text{нечетное} \\ 2 \cdot 4 \cdot 6 \cdot \dots \cdot n, & \text{если } n - \text{четное} \end{cases} \right.$$

2. Вычислите $\max(\min(3,5), \min(2,6))$.
3. Опишите функцию $\text{Stepen}(x,n)$, зависящую от вещественного x и натурального n и вычисляющую (посредством умножения) величину x^n , и используйте её для вычисления значения выражения $2 \cdot 7^k + (a+1)^{-5}$.
4. Даны три натуральных числа. Определите их наибольший общий делитель.
5. Опишите процедуру $\text{Socr}(a,b,p,q)$, зависящую от целых параметров ($b \neq 0$), которая приводит дробь a/b к несократимому виду p/q .

6. Пусть процедура $\text{MaxMin}(x, y)$ присваивает параметру x большее из вещественных чисел x и y , а параметру y — меньшее. Опишите данную процедуру и используйте её для перераспределения значений вещественных переменных a , b и c так, чтобы выполнялось $a \geq b \geq c$.
7. Опишите функцию $F(m, n) = \frac{n! \cdot m!}{(n + m)!}$, где n и m — неотрицательные целые числа.
8. Даны координаты вершин двух треугольников. Определите, какой из них имеет большую площадь.
9. Даны координаты вершин треугольника и координаты некоторой точки внутри него. Найдите расстояние от данной точки до ближайшей стороны треугольника. Указание. При определении расстояний учтите, что площадь треугольника вычисляется и через три его стороны, и через основание и высоту.
10. Дано натуральное число n . Выясните, является ли оно полным квадратом. Определите функцию, позволяющую распознавать полные квадраты.

Критерии оценки:

- оценка «зачтено» выставляется студенту, если он правильно решил 8 или более задач;
- оценка «не зачтено» выставляется студенту, если он правильно решил менее 8 задач.

19.3.3 Задачи к Лабораторной работе №3

1. На вход подаются четыре целых числа; каждое число вводится с новой строки. Вывести минимальное из них.
2. На вход в первой строке подаются два целых положительных числа M и N и вещественное число D , а во второй строке — набор из N вещественных чисел. Сформировать и вывести двумерный вещественный массив размера $M \times N$, у которого первая строка совпадает с исходным набором чисел, а элементы каждой следующей строки равны соответствующему элементу предыдущей строки, умноженному на D (в результате каждый столбец массива будет содержать элементы геометрической прогрессии). Каждую строку элементов массива выводить на новой экранной строке, для каждого числа отводить 8 экранных позиций.
3. На вход в первой строке подается целое число A , во второй строке — целое положительное число N , а в третьей строке — массив из N целых чисел. Найти номер последнего элемента массива со значением A (элементы нумеруются от 1). Если требуемые элементы в массиве отсутствуют, то вывести 0.
4. На вход в первой строке подается целое положительное число N , а во второй строке — массив из N целых чисел. Выполнив однократный просмотр массива, найти максимальное количество подряд идущих одинаковых элементов.
5. На вход в первой строке подается целое число A , во второй строке — целые положительные числа M и N , а в последующих M строках — двумерный целочисленный массив размера $M \times N$. Найти количество строк массива, содержащих элемент со значением A .
6. На вход в первой строке подаются целые положительные числа $N1$ и $N2$, во второй строке — массив из $N1$ целых чисел, а в третьей строке — массив из $N2$ целых чисел. Элементы обоих массивов упорядочены по возрастанию. Выполнить слияние элементов исходных массивов в результирующий массив из $N1 + N2$ чисел, сохранив упорядоченность элементов. Алгоритм сортировки не использовать. Вывести результирующий массив в одной строке, для каждого элемента отводить 4 экранные позиции.

7. На вход в первой строке подается символ C — заглавная латинская буква, а во второй строке — строка S с текстом на английском языке, оканчивающаяся точкой. Найти частоту появления в строке S буквы C без учета ее регистра (частота является вещественным числом и определяется как отношение числа вхождений прописных и строчных букв C к общему числу символов в строке).
8. На вход в первой строке подается целое положительное число N , а во второй строке — массив из N неотрицательных целых чисел. Выполнив однократный просмотр массива, вывести одно из трех чисел: максимальный двузначный элемент, если двузначных элементов в массиве больше, чем трехзначных; минимальный трехзначный элемент массива, если трехзначных элементов в массиве больше, чем двузначных; число 0, если массив содержит одинаковое количество двузначных и трехзначных элементов.
9. На вход в первой строке подается целое четное положительное число N , а во второй строке — массив из N неотрицательных целых чисел. Найти номер первого из двух последовательных элементов, сумма которых максимальна (если таких пар несколько, то надо выбрать первую из них). Элементы нумеруются от 1.
10. Дан целочисленный массив из 15 элементов. Элементы массива могут принимать значения от 30 до 50 — количество яблок в одном ящике. Среди ящиков, содержащих более 40 яблок, найти номер ящика с самыми крупными яблоками. Гарантируется, что такой ящик один. Суммарный вес яблок во всех ящиках одинаковый, ящики нумеруются от 1.

Критерии оценки:

- оценка «зачтено» выставляется студенту, если правильно или с незначительными недочётами решены все задачи;
- оценка «не зачтено» выставляется студенту, решены не все задачи или в решениях имеются серьёзные недочёты.

19.3.4 Контрольные вопросы к Модулю №1 Лабораторного практикума

1. Чем отличаются статические и динамические величины?
2. Что такое указатель?
3. Какие виды указателей вам известны?
4. Приведите примеры объявления указателей.
5. Как выделить память под динамическую переменную?
6. Как освободить память от динамической переменной?
7. Дайте характеристику подпрограмм для работы с динамическими данными (GetMem, FreeMem, Seg, Of, Ptr, @, Mark, Release).
8. Что в языке Pascal обозначает константа nil?
9. Какие ситуации приводят к возникновению в динамически распределяемой памяти «мусора»?
10. Что понимают под «связанным списком»?
11. Какие основные действия над списками и компонентами списков обычно реализуют?
12. Как описывается список?

Критерии оценки:

- оценка «зачтено» выставляется студенту, если студент хорошо ориентируется в теоретическом материале, умеет применять теоретические сведения для решения задач на динамические структуры данных;
- оценка «не зачтено» выставляется студенту, если студент испытывает затруднения при ответе на теоретические вопросы, не способен решать типовые задачи на динамические структуры данных.

19.3.5 Контрольные вопросы к Модулю №2 Лабораторного практикума

1. Какую структуру данных называют очередью? Что такое хвост и голова очереди?
2. На базе каких структур данных может быть организована очередь?
3. Приведите из жизни примеры организации чего-либо по принципу очереди.
4. Какую структуру данных называют деком? В чём его отличия от очереди?
5. Какую структуру данных называют стеком?
6. На базе каких структур может быть организован стек?
7. Приведите из жизни примеры организации чего-либо по принципу стека

Критерии оценки:

- оценка «зачтено» выставляется студенту, если студент хорошо ориентируется в теоретическом материале, умеет применять теоретические сведения для решения задач на динамические структуры данных;
- оценка «не зачтено» выставляется студенту, если студент испытывает затруднения при ответе на теоретические вопросы, не способен решать типовые задачи на динамические структуры данных.

19.3.6 Контрольные вопросы к Модулю №3 Лабораторного практикума

1. Каков принцип построения динамической структуры «дерево»?
2. Перечислите сходства и отличия динамических структур типа «линейный список», «стек», «дерево».
3. Перечислите структуры, которые можно представить в виде дерева, встречающиеся в повседневной жизни.
4. Закончите фразу: «Линейный список — это дерево, в котором ...»

Критерии оценки:

- оценка «зачтено» выставляется студенту, если студент хорошо ориентируется в теоретическом материале, умеет применять теоретические сведения для решения задач на динамические структуры данных;
- оценка «не зачтено» выставляется студенту, если студент испытывает затруднения при ответе на теоретические вопросы, не способен решать типовые задачи на динамические структуры данных.

19.3.7 Контрольные вопросы к Лабораторной работе «Основы Lazarus»

1. Опишите назначение окон интегрированной среды разработки Lazarus.
2. Как организована программа в Lazarus?
3. Какие файлы входят в состав проекта Lazarus?
4. Что такое события? Обработчик события?
5. Как обрабатываются исключительные ситуации в Lazarus?
6. Опишите работу с компонентами TLabel, TButton, TEdit, TMemo.
7. Опишите работу с компонентами TRadioGroup, TComboBox.
8. Опишите работу с компонентом TTimer.
9. Опишите работу с компонентом TChart.
10. Опишите работу с компонентом TStringGrid.
11. Как использовать стандартные диалоги в Lazarus.
12. Расскажите о принципах создания графики в Lazarus с помощью класса TCanvas

Критерии оценки:

- оценка «зачтено» выставляется студенту, если студент хорошо ориентируется в теоретическом материале, умеет применять теоретические сведения для создания сетевых ресурсов;
- оценка «зачтено» выставляется студенту, если студент хорошо ориентируется в теоретическом материале, умеет применять теоретические сведения для программ в ИСР Lazarus;

- оценка «не зачтено» выставляется студенту, если студент испытывает затруднения при ответе на теоретические вопросы, не способен создавать программы в ИСП Lazarus.

19.3.8 Темы индивидуальных проектов

Задание на выполнение индивидуального мини-проекта

С использованием среды программирования Lazarus создать программу из списка. Программа должна обладать удобным пользовательским интерфейсом.

Темы проектов

1. Аналог «Блокнота».
2. Простейший графический редактор (рисование «карандашом» разными цветами) (за рисование фигур – дополнительные баллы).
3. Аналог калькулятора Windows (за «инженерный» – дополнительные баллы).
4. Простейший аудио-плеер со списком воспроизведения.
5. Тестовая оболочка.
6. Браузер.
7. Электронные таблицы.
8. Тетрис.
9. Морской бой.
10. Крестики-нолики.
11. Шашки.
12. Сапер.
13. Пятнашки.
14. Пинг-понг.

Критерии оценки:

- Оценивается удобство и качество пользовательского интерфейса приложения, а также использованные алгоритмы и качество исходного кода.

19.3.9 Вопросы к экзамену

1. Понятие динамических структур данных. Статические и динамические данные.
2. Указатели в Pascal. Объявление. Работа с указателями.
3. Линейный односвязный список. Объявление. Создание. Просмотр. Добавление элементов: в начало списка, в конец списка, после заданного элемента списка.
4. Линейный односвязный список. Объявление. Создание. Просмотр. Удаление элементов списка: первого, последнего, следующего после заданного элемента списка.
5. Очередь. Дек. Стек. Реализация динамической структуры данных стек. Операции со стеком. Алгоритм проверки правильности расстановки скобок.
6. Формы записи арифметических выражений. Постфиксная форма записи выражения. Алгоритм вычисления арифметических выражений, записанных в постфиксной форме.
7. Формы записи арифметических выражений. Постфиксная форма записи выражения. Алгоритм перевода из инфиксной формы в постфиксную.
8. Граф. Дерево. Двоичные деревья. Объявление. Создание двоичного дерева. Способы обхода дерева.
9. Объектно-ориентированное программирование. Классы. Поля. Методы. Свойства. Объекты.
10. Базовые принципы объектно-ориентированного программирования.
11. Иерархия базовых классов Lazarus. Описание класса. Директивы для задания областей видимости в описании классов.

12. Интегрированная среда разработки приложений Lazarus. Характеристика среды. Организация программ в Lazarus. Состав проекта Lazarus. Обработка исключительных ситуаций в Lazarus.
13. События в Lazarus. Понятие события. Обработчик события. События клавиатуры и мыши.
14. Палитра компонентов Lazarus. Компоненты TLabel, TButton, TEdit, TMemo.
15. Палитра компонентов Lazarus. Компоненты TRadioGroup, TComboBox, TTimer.
16. Палитра компонентов Lazarus. Компоненты TChart, TStringGrid.
17. Палитра компонентов Lazarus. Использование стандартных диалогов в Lazarus: компоненты панели Dialogs.
18. Графика в Lazarus. Класс TCanvas. Основные свойства и методы.

Критерии оценки:

- оценка «отлично» выставляется студенту, если студент отлично ориентируется в теоретическом материале, умеет применять теоретические сведения для решения стандартных задач, задач повышенной сложности, творческих задач;
- оценка «хорошо» выставляется студенту, если студент хорошо ориентируется в теоретическом материале, умеет применять теоретические сведения для решения стандартных задач и задач повышенной сложности;
- оценка «удовлетворительно» выставляется студенту, выставляется студенту, если студент испытывает затруднения при ответе на теоретические вопросы, умеет применять теоретические сведения для решения стандартных задач;
- оценка «неудовлетворительно» выставляется студенту, если студент имеет серьёзные пробелы в теоретических знаниях, не способен решать стандартные задачи.

19.4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Оценка знаний, умений и навыков, характеризующая этапы формирования компетенций в рамках изучения дисциплины осуществляется в ходе текущей и промежуточной аттестаций.

Текущий контроль успеваемости проводится в соответствии с Положением о текущей аттестации обучающихся по программам высшего образования Воронежского государственного университета. Текущий контроль успеваемости проводится в формах устного опроса (индивидуальный опрос), выполнения лабораторных работ и заданий лабораторного практикума; выполнения индивидуальных проектных (практико-ориентированных) заданий. Критерии оценивания приведены выше. Промежуточная аттестация проводится в соответствии с Положением о промежуточной аттестации обучающихся по программам высшего образования.

Контрольно-измерительные материалы промежуточной аттестации включают в себя теоретические вопросы, позволяющие оценить уровень полученных знаний и практическое задание, позволяющее оценить степень сформированности умений и навыков.

При оценивании используются количественные шкалы оценок. Критерии оценивания приведены выше.