


Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Алейник Станислав Николаевич
Должность: Ректор
Дата подписания: 08.04.2021 18:21:19
Уникальный программный ключ:
5258223550ea9fbeb23726a1609b644b33d8986ab6255891f288f913a1351fae

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ
имени В.Я.ГОРИНА**

«УТВЕРЖДАЮ»


Декан инженерного факультета
С.В. Стребков
«06» 07 2018 г.

РАБОЧАЯ ПРОГРАММА

по дисциплине **«Высокоуровневые методы информатики
и программирования»**

Направление 09.03.03 Прикладная информатика

Направленность (профиль) - Прикладная информатика в АПК

Квалификация – бакалавр

Майский, 2018

Рабочая программа составлена с учетом требований:

- федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.03 «Прикладная информатика», утвержденного приказом Министерства образования и науки РФ от 12 марта 2015 г. № 207;
- порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры утвержденного приказом Министерства образования и науки РФ от 05.04.2017 г. № 301;
- основной профессиональной образовательной программы ФГБОУ ВО Белгородский ГАУ по направлению подготовки 09.03.03 «Прикладная информатика»

Составитель: канд.техн.наук, доцент Игнатенко В.А.

Рассмотрена на заседании кафедры информатики и информационных технологий от *21.06*, 2018 г., протокол № *13*

и.о. зав. кафедрой



Игнатенко В.А.

Одобрена методической комиссией инженерного факультета от *05.07*, 2018 г., протокол № *9-17/18*

Председатель методической комиссии



Слободюк А.П.

1. ЦЕЛЬ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цель дисциплины – сформировать навыки создания программных продуктов с использованием современных средств разработки, изучить технологию использования визуальных средств программирования.

1.2. Задачи:

- изучить основы проектирования программного обеспечения;
- изучить технологии модульного и объектно-ориентированного программирования;
- изучить приемы работы с визуальными средами программирования;
- приобретение навыков разработки и тестирования программных продуктов функционирующих под управлением современных операционных систем.

II. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ (ОПОП)

2.1. Цикл (раздел) ОПОП, к которому относится дисциплина
Дисциплина «Высокоуровневые методы информатики и программирования» относится к дисциплинам вариативной части (Б1.В.ДВ.09.02) основной образовательной программы.

2.2. Логическая взаимосвязь с другими частями ОПОП

Наименование предшествующих дисциплин, практик, на которых базируется данная дисциплина (модуль)	1. Математика
	2. Информатика и программирование
	3. Информационные системы и технологии
Требования к предварительной подготовке обучающихся	<p><i>знать:</i></p> <ul style="list-style-type: none">• основы теории алгоритмов;• основные программные конструкции;• основы концепций функционального и объектно-ориентированного программирования. <p><i>уметь:</i></p> <ul style="list-style-type: none">• реализовывать программные алгоритмы на языке программирования PASCAL, Си. <p><i>владеть:</i></p> <ul style="list-style-type: none">• Интегрированными средами программирования и отладки (IDE);

Дисциплина является предшествующей для таких дисциплин как «Программирование информационных систем», «Прикладное программирование», «Разработка мобильных приложений».

III. ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ

ДИСЦИПЛИНЫ, СООТВЕТСТВУЮЩИЕ ФОРМИРУЕМЫМ КОМПЕТЕНЦИЯМ

Коды компетенций	Формулировка компетенции	Планируемые результаты обучения по дисциплине
ПК-8	способностью программировать приложения и создавать программные прототипы решения прикладных задач	<p>Знать:</p> <ul style="list-style-type: none"> • основные концепции объектно-ориентированного программирования; • технологию быстрой разработки приложений; • понятие и характеристики объекта, типы отношений между объектами; • понятие класса и виды отношений между классами, понятия переменных и операций класса, группирования классов и интерфейса; • методы объектно-ориентированного анализа; • основные приемы объектно-ориентированного программирования.
		<p>Уметь:</p> <ul style="list-style-type: none"> • самостоятельно составлять, отлаживать, программы на языке Object Pascal с использование объектно-ориентированной методологии программирования;
		<p>Владеть:</p> <ul style="list-style-type: none"> • методами и средствами проектирования и реализации программ с объектно-ориентированных языков программирования и разработки программ в объектно-ориентированных средах;
ПК-12	способностью проводить тестирование компонентов программного обеспечения ИС	<p>Знать:</p> <ul style="list-style-type: none"> • технологию быстрой разработки приложений; • основные причины сложности программного обеспечения и пути их преодоления с помощью новых технологий программирования;
		<p>Уметь:</p> <ul style="list-style-type: none"> • самостоятельно отлаживать, тестировать и документировать программы на языке Object Pascal с использование объектно-ориентированной методологии программирования;
		<p>Владеть:</p>

		<ul style="list-style-type: none"> • Методами автоматического и ручного тестирования программного обеспечения
ПК-14	способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач	Знать: <ul style="list-style-type: none"> • современные методы и средства разработки программ с использованием объектно-ориентированного подхода. • Эффективные инструменты взаимодействия пользовательских приложений с базами данных.
		Уметь: <ul style="list-style-type: none"> • Составлять корректные и эффективные обращения приложений пользователя к базе данных.
		Владеть: <ul style="list-style-type: none"> • Средствами проектирования и контроля работы баз данных.

IV. ОБЪЕМ, СТРУКТУРА, СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, ВИДЫ УЧЕБНОЙ РАБОТЫ И ФОРМЫ КОНТРОЛЯ ЗНАНИЙ

4.1. Распределение объема учебной работы по формам обучения

Вид работы	Объем учебной работы, час	
	Очная	Заочная
Формы обучения (вносятся данные по реализуемым формам)	Очная	Заочная
Семестр (курс) изучения дисциплины	4 семестр 2 курс	3 курс
Общая трудоемкость, всего, час	144	144
<i>зачетные единицы</i>	4	4
Контактная работа обучающихся с преподавателем	40	14
Аудиторные занятия (всего)	40	14
В том числе:		
Лекции	20	6
Лабораторные занятия	20	8
Практические занятия	-	-
<i>Иные виды работ в соответствии с учебным планом (учебная практика)</i>	-	-
Внеаудиторная работа (всего)	20	6
В том числе:		
Контроль самостоятельной работы (на 1 подгруппу в форме компьютерного тестирования)	-*	-
Консультации согласно графику кафедры	20	6
<i>Иные виды работ в соответствии с учебным планом (курсовая работа, РГЗ и др.)</i>	-	-
Промежуточная аттестация	4	4
В том числе:		
Зачет	4	4
Экзамен (на 1 группу)	-	-
Консультация предэкзаменационная (на 1 группу)	-	-
Самостоятельная работа обучающихся	80	120
Самостоятельная работа обучающихся (всего)	80	120
в том числе:		
Самостоятельная работа по проработке лекционного материала (60% от объема лекций)	12	4
Самостоятельная работа по подготовке к лабораторно-практическим занятиям (60% от объема аудиторных занятий)	12	5
Работа над темами (вопросами), вынесенными на самостоятельное изучение	46	91
Самостоятельная работа по видам индивидуальных заданий : подготовка реферата (контрольной работы)	10	20

Примечание: *осуществляется на аудиторных занятиях

4.2 Общая структура дисциплины и виды учебной работы

Наименование модулей и разделов дисциплины	Объемы видов учебной работы по формам обучения, час									
	Очная форма обучения					Заочная форма обучения				
	Всего	Лекции	Лабораторно-практич. занятия	Внеаудиторная работа и пр. атт.	Самостоятельная работа	Всего	Лекции	Лабораторно-практич. занятия	Внеаудиторная работа и пр. атт.	Самостоятельная работа
1	2	3	4	5	6	7	8	9	10	11
Модуль 1. «Алгоритмизация и визуальное программирование в среде Pascal на платформе LAZARUS»	65	10	10	10	35	58	2	4	2	50
1. Программирование в Lazarus	7	2	-	Консультации	5	10	-	-	Консультации	10
2. Разработка интерфейса пользователя	11	2	2		7	12	2	-		10
3. Основы визуального программирования	11	2	2		7	12	-	2		10
4. Палитра компонентов	11	2	2		7	12	-	2		10
5. Интеграция со сторонними приложениями	11	2	2		7	10	-	-		10
<i>Итоговое занятие по модулю 1</i>	4	-	2		2	-	-	-		-
Модуль 2. «Классы, диалоги и графическая поддержка»	65	10	10	10	35	62	4	4	4	50
1. Системные компоненты «дата/время»	9	2	2	Консультации	5	10	-	-	Консультации	10
2. Диалоги	11	2	2		7	10	-	-		10
3. Компоненты работы с графикой	11	2	2		7	14	2	2		10
4. Шаблоны классов	9	2	-		7	14	2	2		10
5. Библиотеки	11	2	2		7	10	-	-		10
<i>Итоговое занятие по модулю 2</i>	4	-	2		2	-	-	-		-
Подготовка реферата в форме презентации (контрольной работы)	10	-	-	-	10	20	-	-	-	20
Зачёт	4	-	-	4	-	4	-	-	4	-

4.3 Структура и содержание дисциплины по формам обучения

Наименование модулей и разделов дисциплины	Объемы видов учебной работы по формам обучения, час									
	Очная форма обучения					Заочная форма обучения				
	Всего	Лекции	Лаб.-практ. зан.	Внеаудит. работа	Самост. работа	Всего	Лекции	Лаб.-практ. зан.	Внеаудит. работа	Самост. работа
1	2	3	4	5	6	7	8	9	10	11
Модуль 1. «Алгоритмизация и визуальное программирование в среде Pascal на платформе LAZARUS»	65	10	10	10	35	58	2	4	2	50
<i>1. Программирование в Lazarus</i>	7	2	-	Консультации	5	10	-	-	Консультации	10
1.1. ООП к проектированию и разработке программ: сущность ООП; объектный тип данных; переменные объектного типа; инкапсуляция; наследование; полиморфизм; классы и объекты. Постановки задач на разработку исполнимого кода и архитектура GUI приложений на платформе LAZARUS	7	2	-		5	10	-	-		10
<i>2. Разработка интерфейса пользователя</i>	11	2	2		7	12	2	-		10
2.1. Конструкторы и деструкторы. Особенности программирования в оконных операционных средах. Основные стандартные модули, обеспечивающие работу в оконной операционной среде. Среда разработки, система окон разработки, система меню. Разбиение задач на подзадачи и рекурсия с инициализацией создания меню.	11	2	2		7	12	2	-		10
<i>3. Основы визуального программирования</i>	11	2	2		7	12	-	2		10
3.1. Ветвление алгоритма, динамическое программирование и создание панелей инструментов для инициализации действий. Основы визуального программирования. Размещение нового компонента	11	2	2		7	12	-	2		10
<i>4. Палитра компонентов</i>	11	2	2		7	12	-	2		10
4.1. Использование компонентов. Типы данных и машинные представления. Компоненты ввода/вывода строковых данных. Командные кнопки.	11	2	2		7	12	-	2		10
<i>5. Интеграция со сторонними компонентами</i>	11	2	2		7	10	-	-		10
5.1. Представление табличных данных в MS Word и MS Excel	11	2	2		7	10	-	-		10
<i>Итоговое занятие по модулю 1</i>	4	-	2	2	-	-	-	-		
Модуль 2. «Классы, диалоги и графическая поддержка»	65	10	10	10	35	62	4	4	4	50
<i>1. Системные компоненты «дата/время»</i>	9	2	2	Консультации	5	10	-	-	Консультации	10
1.1. Классы Pascal и компоненты для работы с датами и временными интервалами	9	2	2		5	10	-	-		10
<i>2. Диалоги</i>	11	2	2		7	10	-	-		10
2.1. Использование диалогов. Компоненты, использование компонентов.	11	2	2		7	10	-	-		10
<i>3. Компоненты работы с графикой</i>	11	2	2		7	14	2	2		10
3.1. Компонент PictureBox. Рисование линий и фигур. Заливка фигур. Вывод текста.	11	2	2		7	14	2	2		10
<i>4. Шаблоны классов</i>	9	2	-		7	14	2	2		10

Наименование модулей и разделов дисциплины	Объемы видов учебной работы по формам обучения, час									
	Очная форма обучения					Заочная форма обучения				
	Всего	Лекции	Лаб.практ. зан.	Внеаудит. работа	Самост. работа	Всего	Лекции	Лаб.практ. зан.	Внеаудит. работа	Самост. работа
1	2	3	4	5	6	7	8	9	10	11
4.1. Шаблоны классов и функций и управление проектами.	9	2	-		7	14	2	2		10
5. Библиотеки	11	2	2		7	10	-	-		10
5.1. Библиотека стандартных шаблонов. Работа с проектом в стиле компонентного программирования	11	2	2		7	10	-	-		10
<i>Итоговое занятие по модулю2</i>	4	-	2		2	-	-	-		-
<i>Подготовка реферата в форме презентации (контрольной работы)</i>	10	-	-	-	10	20	-	-	-	20
Зачёт	4	-	-	4	-	4	-	-	4	-

V. ОЦЕНКА ЗНАНИЙ И ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ ЗНАНИЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

5.1. Формы контроля знаний, рейтинговая оценка и формируемые компетенции (дневная форма обучения)

№ п/п	Наименование рейтингов, модулей и блоков	Формируемые компетенции	Объем учебной работы					Форма контроля знаний	Количество баллов (max)
			Общая трудоемкость	Лекции	Лабор.-практ.заня	Внеаудиторн. раб. и промежут. аттест.	Самост. работа		
Всего по дисциплине		ПК-8 ПК-12 ПК-14	144	20	20	24	80	Зачёт	100
<i>I. Входной рейтинг</i>								Устный опрос	5
<i>II. Рубежный рейтинг</i>								Сумма баллов за модули	60
Модуль 1. «Алгоритмизация и визуальное программирование в среде Pascal на платформе LAZARUS»		ПК-8 ПК-12 ПК-14	65	10	10	10	35		30
1.	Основы программирования в Lazarus		7	2	-	<i>Консультации</i>	5	Устный опрос	
2.	Разработка интерфейса пользователя		11	2	2		7	Устный опрос задачи	
3.	Основы визуального программирования		11	2	2		7	Устный опрос задачи	
4.	Палитра компонентов		11	2	2		7	Устный опрос задачи	
5.	Интеграция со сторонними приложениями		11	2	2		7	Устный опрос задачи	
Итоговый контроль знаний по темам модуля 1.			4	-	2		2	Тестирование	
Модуль 2. «Классы, диалоги и графическая поддержка»		ПК-8 ПК-12 ПК-14	65	10	10	10	35		30
1.	Системные компоненты «дата/время»		9	2	2	<i>Консультации</i>	5	Устный опрос задачи	
2.	Диалоги		11	2	2		7	Устный опрос задачи	
3.	Компоненты работы с графикой		11	2	2		7	Устный опрос задачи	
4.	Шаблоны классов		9	2	-		7	Устный опрос задачи	
5.	Библиотеки		11	2	2		7	Устный опрос задачи	

Итоговый контроль знаний по темам модуля 2.		4	-	2		2	Тестирование,	
<i>III. Творческий рейтинг</i>		10	-	-	-	10	Реферат	5
<i>IV. Выходной рейтинг</i>		4	-	-	4	-	<i>Зачёт</i>	30

5.2. Оценка знаний студента

5.2.1. Основные принципы рейтинговой оценки знаний

Уровень развития компетенций оценивается с помощью рейтинговых баллов.

Рейтинги	Характеристика рейтингов	Максимум баллов
Входной	Отражает степень подготовленности студента к изучению дисциплины. Определяется по итогам входного контроля знаний на первом практическом занятии.	5
Рубежный	Отражает работу студента на протяжении всего периода изучения дисциплины. Определяется суммой баллов, которые студент получит по результатам изучения каждого модуля.	60
Творческий	Результат выполнения студентом индивидуального творческого задания различных уровней сложности, в том числе, участие в различных конференциях и конкурсах на протяжении всего курса изучения дисциплины.	5
Выходной	Является результатом аттестации на окончательном этапе изучения дисциплины по итогам сдачи экзамена. Отражает уровень освоения информационно-теоретического компонента в целом и основ практической деятельности в частности.	30
Общий рейтинг	Определяется путём суммирования всех рейтингов	100

Итоговая оценка компетенций студента осуществляется путём автоматического перевода баллов общего рейтинга в стандартные оценки.

Не зачтено	Зачтено
менее 60 балла	60-100 баллов

5.2.2. Критерии оценки знаний студента на зачете

Оценка на зачете определяется на основании следующих критериев:

- оценка «зачтено» ставится студенту, показавшему систематическое и достаточно глубокое знание учебного материала, умение свободно выполнять ситуационные и тестовые задания, предусмотренные программой, умение логически мыслить и формулировать свою позицию по проблемным вопросам. Зачет может получить студент, который правильно ответил на теоретические вопросы, допустив при этом недочеты непринципиального характера и правильно решившему предложенную на зачете задачу.

- оценка «не зачтено» ставится студенту, обнаружившему существенные пробелы в знании основного учебно-программного материала, допустившему

принципиальные ошибки в выполнении предусмотренных программой заданий.

5.3. Фонд оценочных средств. Типовые контрольные задания или иные материалы, необходимые для оценки формируемых компетенций по дисциплине (приложение 2)

VI. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Основная учебная литература

1. Шакин, В.Н. Объектно-ориентированное программирование на Visual Basic в среде Visual Studio .Net [Электронный ресурс] /В.Н.Шакин, А.В.Загвоздкина, Г.К.Сосновиков - М.: Форум,ИНФРА-М, 2015. - 400 с. Режим доступа: <http://znanium.com/bookread2.php?book=501448>

6.2. Дополнительная литература

1. Немцова, Т.И. Программирование на языке высокого уровня. Программирование на языке Object Pascal: Учебное пособие [Электронный ресурс] / Т.И. Немцова; Под ред. Л.Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2015. - 496 с. Режим доступа: <http://znanium.com/bookread2.php?book=472870>

6.3. Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине

Самостоятельная работа студентов заключается в инициативном поиске информации о наиболее актуальных проблемах, которые имеют большое практическое значение и являются предметом научных дискуссий в рамках изучаемой дисциплины.

Самостоятельная работа планируется в соответствии с календарными планами рабочей программы по дисциплине и в методическом единстве с тематикой учебных аудиторных занятий.

Самостоятельную работу студента поддерживает электронная информационная среда ВУЗа, доступ к которой <http://do.belgau.edu.ru> (логин, пароль студента)

6.3.1. Методические указания по освоению дисциплины

Игнатенко, В.А. Методические указания и задания к выполнению лабораторно-практических и самостоятельных работ по дисциплине "Высокоуровневые методы информатики и программирования" для студентов экономического факультета направления "Прикладная информатика" [Электронный ресурс]: методические указания / Белгородский ГАУ; сост. В. А. Игнатенко. - Белгород: Белгородский ГАУ, 2014. - 48 с. Режим доступа: <https://clck.ru/FDrZr>

6.3.2. Видеоматериалы

1. <https://www.youtube.com/watch?v=UBbE0p7oU-U&list=PLaHMNOpHDYwpfIA6-ykJrkng0pd2smyeg>
2. https://www.youtube.com/watch?v=N_PZVx062bo
3. https://www.youtube.com/watch?v=u39itLZw5_8&list=PLdVuyun26swWqJe3IGKWjbPYsDMfeJ9KY

6.3.3 Печатные периодические издания

1. Журнал «Информационные технологии»
2. Журнал «Моделирование и анализ информационных систем»
3. Журнал «Вестник российской сельскохозяйственной науки»
4. Журнал «Достижения науки и техники АПК»
5. Журнал «Экономика, статистика и информатика»

6.4. Ресурсы информационно-телекоммуникационной сети «Интернет», современные профессиональные базы данных, информационные справочные системы.

1. Профессиональная база данных и информационно справочная система по официальной технической документации для разработчиков под ОС Microsoft Windows <https://msdn.microsoft.com/ru-ru>
2. Профессиональная база данных и информационно справочная система по официальной технической документации для разработчиков под ОС Microsoft Windows <https://technet.microsoft.com/ru-ru>
3. Профессиональная база данных стандартов <http://iso.gost.ru/wps/portal/>
4. Профессиональная база данных и информационно справочная система по официальной технической документации Lazarus <https://www.lazarus-ide.org/>

6.5. Перечень программного обеспечения, информационных технологий.

1. Операционная система Windows;
2. Пакет программ Microsoft Office;
3. SunRav- программа для тестирования;
4. Lazarus (бесплатная визуальная среда разработки на основе Free Pascal).

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для преподавания дисциплины используются:

1. учебная аудитория лекционного типа, оборудованная мультимедийным оборудованием для демонстрации презентаций;

2. компьютерный класс для проведения лабораторно – практических занятий.

3. помещение для самостоятельной работы обучающихся, оснащенное компьютерной техникой с подключением к сети Интернет и электронной информационно-образовательной среде ВУЗа.

4. Лаборатория прикладной информатики и информационных технологий

VIII. ПРИЛОЖЕНИЯ

Приложение 1

СВЕДЕНИЯ О ДОПОЛНЕНИИ И ИЗМЕНЕНИИ РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ НА 201 / 201 УЧЕБНЫЙ ГОД

Высокоуровневые методы информатики и программирования

дисциплина (модуль)

09.03.03 «Прикладная информатика»

направление подготовки/специальность

ДОПОЛНЕНО (с указанием раздела РПД)

ИЗМЕНЕНО (с указанием раздела РПД)

УДАЛЕНО (с указанием раздела РПД)

Реквизиты протоколов заседаний кафедр, на которых пересматривалась программа

Кафедра информатики и информационных технологий	Кафедра информатики и информационных технологий
от _____ № _____ Дата	от _____ № _____ дата

Методическая комиссия инженерный факультета

« ___ » _____ 201_ года, протокол № _____

Председатель методкомиссии _____ Слободюк А. П.

Декан инженерный факультета

Стребков С.В.

« ___ » _____ 201_ г

Согласовано:

Генеральный директор
ООО «Вектор Программные Системы»
« 05 » мая 2018 г.
Кочетов В.А.



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения промежуточной аттестации обучающихся
по дисциплине **Высокоуровневые методы информатики и
программирования**
Направление подготовки **09.03.03 Прикладная информатика**
Профиль **«Прикладная информатика в АПК»**

Майский, 2018

1. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Код контролируемой компетенции	Формулировка контролируемой компетенции	Этап (уровень) освоения компетенции	Планируемые результаты обучения	Наименование модулей и (или) разделов дисциплины	Наименование оценочного средства	
					Текущий контроль	Промежуточная аттестация
ПК-8	способность программировать приложения и создавать программные прототипы решения прикладных задач	Первый этап (пороговой уровень)	знать: 1) основные концепции объектно-ориентированного программирования; 2) технологию быстрой разработки приложений; 3) понятие и характеристики объекта, типы отношений между объектами; 4) понятие класса и виды отношений между классами, понятия переменных и операций класса, группирования классов и интерфейса; 5) методы объектно-ориентированного анализа; 6) основные приемы объектно-ориентированного программирования.	Модуль 1	устный опрос	итоговое тестирование, вопросы к зачёту
					подготовка рефератов	
					тестирование	
				Модуль 2	устный опрос	итоговое тестирование, вопросы к зачёту
		подготовка рефератов				
		тестирование, ситуационные задачи				
Второй этап (продвинутый уровень)	уметь: 1) самостоятельно составлять, отлаживать, программы на языке Object Pascal с использование объектно-ориентированной методологии программирования	Модуль 1	устный опрос	итоговое тестирование, вопросы к зачёту		
			подготовка рефератов			
			тестирование, ситуационные задачи			

				Модуль 2	устный опрос	итоговое тестирование, вопросы к зачёту
					подготовка рефератов	
					тестирование, ситуационные задачи	
		Третий этап (высокий уровень)	владеть: 1) методами и средствами проектирования и реализации программ с объектно-ориентированных языков программирования и разработки программ в объектно-ориентированных средах	Модуль 1	устный опрос	итоговое тестирование, вопросы к зачёту
					подготовка рефератов	
					тестирование, ситуационные задачи	
				Модуль 2	устный опрос	итоговое тестирование, вопросы к зачёту
					подготовка рефератов	
					тестирование, ситуационные задачи	
ПК-12	Способность проводить тестирование компонентов программного обеспечения ИС	Первый этап (пороговой уровень)	знать: 1) технологию быстрой разработки приложений; 2) основные причины сложности программного обеспечения и пути их преодоления с помощью новых технологий программирования.	Модуль 1	устный опрос	итоговое тестирование, вопросы к зачёту
					подготовка рефератов	
					тестирование, ситуационные задачи	
				Модуль 2	устный опрос	
			подготовка рефератов			
	тестирование, ситуационные задачи					
	Второй этап	уметь:	Модуль 1	устный опрос	итоговое	

		(продвинуты й уровень)	1) самостоятельно отлаживать, тестировать и документировать программы на языке Object Pascal с использованием объектно-ориентированной методологии программирования.		подготовка рефератов	тестирование, вопросы к зачёту
					тестирование, ситуационные задачи	
		Третий этап (высокий уровень)	владеть: 1) Методами автоматического и ручного тестирования программного обеспечения.	Модуль 2	устный опрос	итоговое тестирование, вопросы к зачёту
					подготовка рефератов	
					тестирование, ситуационные задачи	
				Модуль 1	устный опрос	итоговое тестирование, вопросы к зачёту
подготовка рефератов						
тестирование, ситуационные задачи						
Модуль 2	устный опрос	итоговое тестирование, вопросы к зачёту				
	подготовка рефератов					
	тестирование, ситуационные задачи					
ПК-14	способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач	Первый этап (пороговой уровень)	знать: 1) современные методы и средства разработки программ с использованием объектно-ориентированного подхода. 2) Эффективные инструменты взаимодействия пользовательских приложений с базами данных;	Модуль 1	устный опрос	итоговое тестирование, вопросы к зачёту
					подготовка рефератов	
					тестирование, ситуационные задачи	
				Модуль 2	устный опрос	итоговое тестирование, вопросы к
					подготовка рефератов	

					тестирование, ситуационные задачи	зачёту
	Второй этап (продвинутый уровень)	уметь: 1) составлять корректные и эффективные обращения приложений пользователя к базе данных	Модуль 1	устный опрос	итоговое тестирование, вопросы к зачёту	
подготовка рефератов						
тестирование, ситуационные задачи						
			Модуль 2	устный опрос	итоговое тестирование, вопросы к зачёту	
подготовка рефератов						
тестирование, ситуационные задачи						
	Третий этап (высокий уровень)	владеть: 1) средствами проектирования и контроля работы баз данных.	Модуль 1	устный опрос	итоговое тестирование, вопросы к зачёту	
подготовка рефератов						
тестирование, ситуационные задачи						
			Модуль 2	устный опрос	итоговое тестирование, вопросы к зачёту	
подготовка рефератов						
тестирование, ситуационные задачи						

2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Компетенция	Планируемые результаты обучения (показатели достижения заданного уровня компетенции)	Уровни и критерии оценивания результатов обучения, шкалы оценивания			
		<i>Компетентность не сформирована</i>	<i>Пороговый уровень компетентности</i>	<i>Продвинутый уровень компетентности</i>	<i>Высокий уровень</i>
		<i>не зачтено</i>	<i>зачтено</i>	<i>зачтено</i>	<i>Зачтено</i>
<i>ПК-8</i>	<i>способность программировать приложения и создавать программные прототипы решения прикладных задач</i>	<i>способность программировать приложения и создавать программные прототипы решения прикладных задач не сформирована</i>	<i>Частично владеет способностью программировать приложения и создавать программные прототипы решения прикладных задач</i>	<i>Владеет способностью программировать приложения и создавать программные прототипы решения прикладных задач</i>	<i>Свободно владеет способностью программировать приложения и создавать программные прототипы решения прикладных задач</i>
	Знать: 1) основные концепции объектно-ориентированного программирования; 2) технологию быстрой разработки приложений; 3) понятие и характеристики объекта, типы отношений между объектами; 4) понятие класса и виды отношений между классами, понятия переменных и операций класса, группирования классов и интерфейса; 5) методы объектно-ориентированного анализа; 6) основные приемы объектно-ориентированного	Допускает грубые ошибки при воспроизведении основных концепций объектно-ориентированного программирования; технологий быстрой разработки приложений; понятий и характеристик объекта, типов отношений между объектами; понятий класса и видов отношений между классами, понятий переменных и операций класса, группирования классов и интерфейса;	Может изложить основные концепции объектно-ориентированного программирования; технологию быстрой разработки приложений; понятие и характеристики объекта, типы отношений между объектами; понятие класса и виды отношений между классами, понятия переменных	Знает основные концепции объектно-ориентированного программирования; технологию быстрой разработки приложений; понятие и характеристики объекта, типы отношений между объектами; понятие класса и виды отношений между классами, понятия переменных и операций класса, группирования классов и интерфейса;	Аргументировано проводит сравнение основных концепций объектно-ориентированного программирования; технологий быстрой разработки приложений; понятий и характеристик объекта, типов отношений между объектами; понятий класса и видов отношений между классами, понятий переменных и операций класса,

	программирования.	методов объектно-ориентированного анализа; основных приемов объектно-ориентированного программирования.	и операций класса, группирования классов и интерфейса; методы объектно-ориентированного анализа; основные приемы объектно-ориентированного программирования.	методы объектно-ориентированного анализа; основные приемы объектно-ориентированного программирования.	группирования классов и интерфейса; методов объектно-ориентированного анализа; основных приемов объектно-ориентированного программирования.
	Уметь: 1) самостоятельно составлять, отлаживать, программы на языке Object Pascal с использованием объектно-ориентированной методологии программирования;	Не умеет самостоятельно составлять, отлаживать, программы на языке Object Pascal с использованием объектно-ориентированной методологии программирования.	Частично умеет самостоятельно составлять, отлаживать, программы на языке Object Pascal с использованием объектно-ориентированной методологии программирования.	Способен самостоятельно составлять, отлаживать, программы на языке Object Pascal с использованием объектно-ориентированной методологии программирования.	Способен самостоятельно организовывать работу по самостоятельному составлению, отладке, программы на языке Object Pascal с использованием объектно-ориентированной методологии программирования.
	Владеть: 1) методами и средствами проектирования и реализации программ с объектно-ориентированных языков программирования и разработки программ в объектно-ориентированных средах;	Не владеет методами и средствами проектирования и реализации программ с объектно-ориентированных языков программирования и разработки программ в объектно-ориентированных	Частично владеет методами и средствами проектирования и реализации программ с объектно-ориентированных языков программирования и разработки программ в объектно-ориентированных	Владеет методами и средствами проектирования и реализации программ с объектно-ориентированных языков программирования и разработки программ в объектно-ориентированных	Свободно владеет методами и средствами проектирования и реализации программ с объектно-ориентированных языков программирования и разработки программ в объектно-

		средах.	средах.	средах.	ориентированных средах.
ПК-12	<i>способностью проводить тестирование компонентов программного обеспечения ИС</i>	<i>способность проводить тестирование компонентов программного обеспечения ИС не сформирована</i>	<i>Частично владеет способностью проводить тестирование компонентов программного обеспечения ИС</i>	<i>Владеет способностью проводить тестирование компонентов программного обеспечения ИС</i>	<i>Свободно владеет способностью проводить тестирование компонентов программного обеспечения ИС</i>
	Знать: 1) технологию быстрой разработки приложений; 2) основные причины сложности программного обеспечения и пути их преодоления с помощью новых технологий программирования;	Допускает грубые ошибки при воспроизведении технологии быстрой разработки приложений; основных причин сложности программного обеспечения и пути их преодоления с помощью новых технологий программирования;	Может изложить технологию быстрой разработки приложений; основные причины сложности программного обеспечения и пути их преодоления с помощью новых технологий программирования.	Знает основы технологии быстрой разработки приложений; основных причин сложности программного обеспечения и пути их преодоления с помощью новых технологий программирования;.	Аргументировано проводит анализ технологии быстрой разработки приложений; основных причин сложности программного обеспечения и пути их преодоления с помощью новых технологий программирования;.
	Уметь: 1) самостоятельно отлаживать, тестировать и документировать программы на языке Object Pascal с использованием объектно-ориентированной методологии программирования;	Не умеет самостоятельно отлаживать, тестировать и документировать программы на языке Object Pascal с использованием объектно-ориентированной методологии программирования.	Частично умеет самостоятельно отлаживать, тестировать и документировать программы на языке Object Pascal с использованием объектно-ориентированной методологии программирования.	Способен самостоятельно отлаживать, тестировать и документировать программы на языке Object Pascal с использованием объектно-ориентированной методологии программирования.	Способен самостоятельно организовывать работу по отладке, тестированию и документированию программы на языке Object Pascal с использованием объектно-ориентированной методологии программирования.
	Владеть:	Не владеет методами	Частично владеет	Владеет методами	Свободно владеет

	1) Методами автоматического и ручного тестирования программного обеспечения	автоматического и ручного тестирования программного обеспечения.	методами автоматического и ручного тестирования программного обеспечения..	автоматического и ручного тестирования программного обеспечения.	методами автоматического и ручного тестирования программного обеспечения.
ПК-14	способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач	<i>способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач не сформирована</i>	<i>Частично владеет способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач</i>	<i>Владеет способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач</i>	<i>Свободно владеет способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач</i>
	Знать: 1) современные методы и средства разработки программ с использованием объектно-ориентированного подхода. 2)Эффективные инструменты взаимодействия пользовательских приложений с базами данных.	Допускает грубые ошибки при воспроизведении современных методов и средств разработки программ с использованием объектно-ориентированного подхода; эффективных инструментов взаимодействия пользовательских приложений с базами данных.	Может изложить современные методы и средства разработки программ с использованием объектно-ориентированного подхода; эффективные инструменты взаимодействия пользовательских приложений с базами данных.	Знает основные современные методы и средства разработки программ с использованием объектно-ориентированного подхода; эффективные инструменты взаимодействия пользовательских приложений с базами данных..	Аргументировано проводит выбор современных методов и средств разработки программ с использованием объектно-ориентированного подхода; эффективных инструментов взаимодействия пользовательских приложений с базами данных..

	<p>Уметь: 1) Составлять корректные и эффективные обращения приложений пользователя к базе данных.</p>	<p>Не умеет составлять корректные и эффективные обращения приложений пользователя к базе данных.</p>	<p>Частично умеет составлять корректные и эффективные обращения приложений пользователя к базе данных..</p>	<p>Способен составлять корректные и эффективные обращения приложений пользователя к базе данных..</p>	<p>Способен самостоятельно организовывать работу по составлению корректных и эффективных обращений приложений пользователя к базе данных..</p>
	<p>Владеть: 1) Средствами проектирования и контроля работы баз данных.</p>	<p>Не владеет средствами проектирования и контроля работы баз данных.</p>	<p>Частично владеет средствами проектирования и контроля работы баз данных.</p>	<p>Владеет средствами проектирования и контроля работы баз данных.</p>	<p>Свободно владеет средствами проектирования и контроля работы баз данных.</p>

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

1.1. Первый этап (пороговой уровень)

ЗНАТЬ (помнить и понимать): студент помнит, понимает и может продемонстрировать широкий спектр фактических, концептуальных, процедурных знаний.

3.1.1. Перечень вопросов для определения входного рейтинга

1. Базовые типы данных. Формы представления констант. Операторы присваивания.
2. Массивы. Арифметические операторы: +, -, *, /, %, ++, --.
3. Структуры. Перечисляемый тип и объединения.
4. Условный оператор и множественный выбор (оператор switch).
5. Циклы: while, do, for. Операция запятая. Досрочное завершение итерации и цикла.
6. Передача параметров при вызове функции. Указатели.
7. Указатели: на структуру, функцию, массив из 10 элементов. Получение адреса.
8. Операция раскрытия указателя. Два способа ссылки на элемент структуры по указателю.
9. Стандартные функции ввода/вывода.
10. Чтение/запись в файл.
11. Глобальные и локальные переменные. Переменные типа auto, static, extern.
12. Отличия в инициализации автоматических и статических переменных.
13. Динамическое распределение памяти. Функции malloc, calloc, free. Оператор sizeof.
14. Массивы и указатели.
15. Истина и ложь в Си. Логические операторы.
16. Битовые операторы: &, |, ^, ~, >>, <<. Примеры.
17. Битовые операторы: Напишите программу, выводящую на экран значение переменной типа unsigned char в двоичном виде.
18. Строка в Си. Напишите программу копирования одной строки в другую, не используя стандартные функции.
19. Директивы препроцессора #include, #define, #define с параметрами, #if, #else, #endif, #ifdef, #ifndef.

3.1.2. Перечень вопросов к зачету

1. Оценка эффективности программы в зависимости от исполняемого алгоритма.
2. Размер задачи как характеристика объема входных данных.
3. Преобразование структуры программы на базе стратегии алгоритма.
4. Основные компоненты среды программирования Lazarus.
5. Структура проекта приложения в Lazarus.
6. Размер задачи как характеристика объема входных данных; временная и емкостная сложность программы как функции размера задачи; верхняя, нижняя и средняя оценки.
7. Разбиение задачи на несколько подзадач меньшего размера.
8. Рекурсия как общий метод сведения задачи к самой себе.

9. Динамическое программирование.
10. Методы простого поиска в массиве.
11. Стандартные типы данных.
12. Простые типы данных языка Object Pascal.
13. Ограничения на основные типы данных.
14. Способы конструирования и верификации программ.
15. Методика визуального программирования.
16. Основные отличия Lazarus и Delphi 7.
17. Обращение к полям структур для объектов и к полям по указателю на объект типа структура.
18. Операции над объектами различных типов данных.
19. Компонент выпадающего списка.
20. Списки с флажками.
21. Главное и контекстное меню.
22. Создание главного меню приложения.
23. Инициализация действий с помощью пунктов меню.
24. Назначение панелей инструментов.
25. Создание панелей инструментов в среде Lazarus.
26. Инициализация действий с помощью кнопок панели инструментов.
27. Назначение строки состояния.
28. Создание строки состояния в среде Lazarus.
29. Отображение подсказки к пунктам меню в строке состояния.
30. Понятие компонента.
31. Иерархия компонентов.
32. Основные свойства и методы компонентов.
33. Статические текстовые компоненты.
34. Текстовые поля ввода.
35. Преобразование текстовых данных в данные других типов.
36. Командные кнопки и их использование.
37. Кнопки (флажки) выбора.
38. Взаимозависимые (радио) кнопки.
39. Назначение списочных компонентов.
40. Компонент выпадающего списка.
41. Списки с флажками.
42. Компонент представления древовидных структур данных (иерархий).
43. Компонент представления двухуровневых иерархий.
44. Методика совместного использования компонентов представления иерархий.
45. Компонент табличного представления данных.
46. Работа с таблицей в целом.
47. Работа с ячейками таблицы.
48. Класс для работы с хронологическими датами.
49. Компонент для работы с хронологическими датами.
50. Класс для работы с временными интервалами.
51. Использование немодальных и модальных форм (диалогов).
52. Использование стандартных диалогов.
53. Стандартные диалоги для открытия и записи файлов.
54. Концепция графического вывода в Windows.
55. Концепция графического вывода в LAZARUS.
56. Библиотеки графических компонентов в LAZARUS.
57. Вывод графики на форму.
58. Событие Paint.
59. Специализированный компонент для вывода графики PictureBox.

60. Компонент пера и методика его использования.
61. Класс Graphic и его система координат.
62. Рисование линий и замкнутых фигур.
63. Стандартные замкнутые фигуры.
64. Использование кистей.
65. Методы заливки плоских фигур.
66. Создание и использование шрифтов.
67. Стандартный диалог для выбора шрифта.
68. Методы вывода текста в графическом компоненте.
69. Шаблоны. Библиотека стандартных шаблонов.
70. Параметризованные классы. Основные свойства шаблонов в классах. Параметризованные функции. Параметры шаблона.
71. Компонентные функции параметризованных классов. Контейнерные классы. Серверы.
72. Использование классов функциональных объектов для настройки шаблонных классов.
73. Управление проектом. Атрибуты и параметры, доступные пользователю.
74. Концептуальное понятие STL. Библиотека стандартных шаблонов STL.
75. Последовательные и ассоциативные контейнеры. Итераторы STL. Общие свойства контейнеров.
76. Использование последовательных контейнеров. Адаптеры контейнеров. Алгоритмы.
77. Шаблоны в компонентном программировании и работа с проектом в стиле объектно-ориентированного программирования.

3.2. Второй этап (продвинутый уровень)





УМЕТЬ (применять, анализировать, оценивать, синтезировать): уметь использовать изученный материал в конкретных условиях и в новых ситуациях; осуществлять декомпозицию объекта на отдельные элементы и описывать то, как они соотносятся с целым, выявлять структуру объекта изучения; оценивать значение того или иного материала – научно-технической информации, исследовательских данных и т. д.; комбинировать элементы так, чтобы получить целое, обладающее новизной


3.2.1. Тестовые задания

1) Как называется окно, в котором отображаются возможные события и свойства объекта?

- a) Дерево объектов;
- b) Инспектор объектов
- c) Форма
- d) Редактор кода

2) Найдите соответствие между компонентами и их названиями:

- | | |
|--|---------------|
| 1.  | a) edit |
| 2.  | b) radiogroup |
| 3.  | c) label |
| 4.  | d) button |

3) что означает компонент 

- a) командная кнопка
 - b) поле ввода и вывода текста
 - c) поле редактирования
 - d) для ввода массива
- 4) Для создания интерфейса будущего приложения служит окно:
- a) Редактор кода
 - b) Форма
 - c) Инспектор объектов
 - d) Дерево объектов
- 5) Для отображения информации о выделенном объекте служит окно:
- a) Редактор кода
 - b) Форма
 - c) Инспектор объектов
 - d) Дерево объектов
- 6) Что означает свойство «Font»
- a) Вид рамки
 - b) Название шрифта
 - c) Цвет шрифта
 - d) Цвет фона
- 7) Что означает свойство «Text»
- a) Признак доступности компонента
 - b) Вид рамки
 - c) Заголовок
 - d) Выводимый или вводимый текст
- 8) Функция перевода строки в целое число:
- a) Inttostr
 - b) Strtoint
 - c) Strtfloat
 - d) Floattostr
- 9) Что означает свойство «Caption»
- a) Размер шрифта
 - b) Высота
 - c) Заголовок
 - d) Ширина
 - e) Вид рамки
- 10) В каком окне создается и редактируется текст программы:
- a) окно формы
 - b) окно редактора
 - c) дерево объектов
 - d) инспектор объектов
- 11) Что означает свойство Borderstyle компонента Edit
- a) ввод текста в поле редактирования
 - b) вид рамки
 - c) вводимый текст
 - d) выводимый текст

- 12) Что означает свойство Wordwrap компонента Label
- a) выравнивание текста
 - b) текст в поле компонента
 - c) перенос слов в длинной надписи
 - d) расстояние от верхней границы
- 13) Компонент Listbox находится в группе
- a) Дополнительно
 - b) Стандартная
 - c) Система
- 14) Компонент, предназначенный для вставки звука
- a) Listbox
 - b) Image
 - c) Timer
 - d) MediaPlayer
- 15) Какие форматы звуковых файлов поддерживает Lazarus
- a) midi, mp3
 - b) wave, mp3
 - c) midi, wave
 - d) wave, mp3

3.2.2. Темы рефератов

1. Сравнение ООП в языках Delphi/Pascal, C++, VB.
2. Динамические объекты.
3. Раннее и позднее связывание.
4. Конструкторы и деструкторы в ООП.
5. Контейнерные классы.
6. Наследование. Множественное наследование.
7. Объекты и сообщения.
8. Простой и сложный полиморфизм.
9. COM в Unisim и HiSys.
10. Стандартные классы ОС Windows.
11. COM, OLE - технологии
12. Операторы языка SQL
13. Создание много уровневых приложений для работы с базами данных
14. Интерфейсы и их реализация в классах
15. Создание и использование COM серверов
16. Возможности приложений MS Office как COM серверов
17. Технология OLE Automation и интерфейсы IDispatch и DispInterface
18. Сокеты в Delphi
19. Основные сервисы Internet
20. Технология WebSnap – распределенные интернет приложения
21. Возможности компонентов INDY
22. Создание Web приложений на основе компонентов IntraWeb
23. Использование данных в формате XML

3.3. Третий этап (высокий уровень)

ВЛАДЕТЬ навыками по применению теоретических и практических знаний и

умений при решении ситуационных задач, практической направленности по дисциплине.

3.3.1. Ситуационные задачи

Задача 1.

На языке Lazarus разработать контейнер «динамический массив» — TArray, который

- содержит элементы типа Pointer,
- увеличивает свой размер при добавлении в него элементов,
- предоставляет типовые свойства (текущее количество элементов, количество зарезервированных элементов),
- предоставляет типовые процедуры, применяемые при работе с массивом (вставка по индексу, удаление по индексу, очистка, копирование в другой массив, поиск индекса элемента по ссылке, сортировка),
- при обращении к элементу массива проверяет, что индекс находится в допустимых пределах.

Продемонстрировать использование класса.

Задача 2.

На языке Lazarus разработать контейнер «коллекция» — TCollection, который:

- содержит элементы типа TCollectionItem, содержащие ссылку на своего владельца,
- увеличивает свой размер при добавлении в него элементов,
- предоставляет типовые свойства (текущее количество элементов, количество зарезервированных элементов),
- предоставляет типовые процедуры, применяемые при работе с коллекцией (вставка, удаление, очистка, копирование в другую коллекцию, поиск индекса элемента по ссылке, сортировка),
- предоставляет типовые события, происходящие при работе с коллекцией (вставка, удаление, очистка),
- при обращении к элементу коллекции проверяет, что индекс находится в допустимых пределах.

Продемонстрировать использование класса.

Задача 3.

На языке Lazarus разработать контейнер «список» — TList, который:

- содержит элементы типа Pointer,
- предоставляет типовые свойства (первый элемент, следующий элемент),
- предоставляет типовые процедуры, применяемые при работе со списком (вставка элемента, удаление элемента, очистка списка, замена элемента, перемещение списка в другой список),
- предоставляет типовые события, происходящие при работе со списком (вставка, удаление, очистка),

Продемонстрировать использование класса.

Задача 4.

На языке Lazarus разработать контейнер «стек» — TStack, который эмулирует стек процессора, и который:

- содержит произвольные данные, представляемые в виде последовательности байтов,
- предоставляет типовые свойства (указатель на вершину стека, указатель на начало «кадра», указатель на дно стека),
- предоставляет типовые процедуры, применяемые при работе со стеком (создание нового «кадра», выталкивание «кадра», загрузка данных заданного размера,

выталкивание данных заданного размера),

- предоставляет типовые события, происходящие при работе со стеком (загрузка в стек, выталкивание из стека),

Продемонстрировать использование класса.

Задача 5.

На языке Lazarus разработать класс «словарь» — TDictionary, который представляет собой набор ключей и их значений, и который:

- содержит ключи типа string и значения типа Pointer,

- увеличивает свой размер при добавлении в него элементов,

- предоставляет типовые свойства (количество ключей и значений, признак сортированного «словаря»),

- предоставляет типовые процедуры (вставка ключа и значения, удаление значения по ключу, поиск значения по ключу, очистка, сортировка),

- предоставляет типовые события, происходящие при работе со «словарем» (вставка, удаление, очистка).

Продемонстрировать использование класса.

Задача 6.

На языке Lazarus разработать класс «XML-дерево» — TXmlTree — для представления XML-файла в памяти. Класс:

- содержит список тегов и значений типа string, а также список вложенных деревьев типа TXmlTree,

- обеспечивает загрузку и выгрузку XML-файла (без детализации),

- предоставляет типовые свойства (количество ключей и значений, количество вложенных деревьев),

- предоставляет типовые процедуры (вставка ключа и значения в заданную позицию, удаление значения по ключу, поиск значения по ключу или на основе «пути» по аналогии с файловой системой, очистка),

- предоставляет типовые события, происходящие при работе с контейнером (вставка, удаление, очистка),

Продемонстрировать использование класса.

Задача 7.

На языке Lazarus разработать абстракцию файла TAbstractFile и два производных класса TOSFile (файл операционной системы) и TMemoryFile (файл в памяти). Классы:

- предоставляют типовые свойства (размер файла, текущая позиция указателя чтения-записи в файле),

- предоставляют типовые процедуры работы с файлом (запись, чтение, перемещение позиции указателя чтения-записи в файле, копирование содержимого другого файла),

- предоставляет типовые события, происходящие при работе с файлом (чтение, запись).

Продемонстрировать использование классов так: программно создать файл на диске с любым содержимым, скопировать содержимое файла на диске в файл в памяти.

Задача 8.

На языке Lazarus разработать контейнер «битовый массив» — TBitArray, который

- содержит элементы типа Boolean так, что на каждый отводится лишь 1 бит,

- увеличивает свой размер при добавлении в него элементов,

- предоставляет типовые свойства (текущее количество элементов, количество зарезервированных элементов),

- предоставляет типовые процедуры, применяемые при работе с массивом (чтение и запись бита по индексу, расширение массива),

- при обращении к биту в массиве проверяет, что индекс находится в допустимых пределах.

Продемонстрировать использование класса.

Задача 9.

На языке Lazarus разработать класс для представления матрицы произвольной размерности – TMatrix. Класс:

- содержит элементы типа Double,
- принимает размеры матрицы на вход конструктора,
- предоставляет типовые свойства (размеры матрицы, элемент на пересечении строки и столбца),
- предоставляет типовые процедуры, применяемые при работе с матрицами (умножение, транспонирование),

Продемонстрировать использование класса.

Задача 10.

На языке Lazarus разработать класс «кольцевой буфер» - TCircularBuffer. Класс:

- содержит байты данных,
- предоставляет типовые свойства (указатель чтения, указатель записи, количество данных в буфере),
- предоставляет типовые процедуры, применяемые при работе с кольцевым буфером (запись, чтение, наращивание размера буфера),
- генерирует исключительную ситуацию при переполнении буфера.

Продемонстрировать использование класса.

Задача 11.

Разработать шаблон «динамический массив» — TArray, который

- содержит элементы заданного типа данных,
- увеличивает свой размер при добавлении в него элементов,
- предоставляет типовые свойства (текущее количество элементов, количество зарезервированных элементов),
- предоставляет типовые процедуры, применяемые при работе с массивом (вставка по индексу, удаление по индексу, очистка, копирование в другой массив, поиск индекса элемента по ссылке, сортировка),
- при обращении к элементу массива проверяет, что индекс находится в допустимых пределах.

Продемонстрировать использование класса.

Задача 12.

Разработать класс «коллекция» — TCollection, который:

- содержит элементы типа TCollectionItem, содержащие ссылку на своего владельца,
- увеличивает свой размер при добавлении в него элементов,
- предоставляет типовые свойства (текущее количество элементов, количество зарезервированных элементов),
- предоставляет типовые процедуры, применяемые при работе с коллекцией (вставка, удаление, очистка, копирование в другую коллекцию, поиск индекса элемента по ссылке, сортировка),
- предоставляет типовые события, происходящие при работе с коллекцией (вставка, удаление, очистка),
- при обращении к элементу коллекции проверяет, что индекс находится в допустимых пределах.

Продемонстрировать использование класса.

Задача 13.

Разработать шаблон списка — TList, который:

- содержит элементы заданного типа данных,
- предоставляет типовые свойства (первый элемент, следующий элемент),
- предоставляет типовые процедуры, применяемые при работе со списком (вставка элемента, удаление элемента, очистка списка, замена элемента, перемещение списка в другой список),
- предоставляет типовые события, происходящие при работе со списком (вставка, удаление, очистка),

Продемонстрировать использование класса.

Задача 14.

Разработать шаблон стека — TStack, который:

- содержит элементы заданного типа данных,
- предоставляет типовые свойства (указатель на вершину стека, указатель на дно стека),
- предоставляет типовые процедуры, применяемые при работе со стеком (загрузка элемента на стек, выталкивание элемента из стека, очистка стека),
- предоставляет типовые события, происходящие при работе со стеком (загрузка в стек, выталкивание из стека),

Продемонстрировать использование класса.

Задача 15.

Разработать шаблон «словарь» — TDictionary, который представляет собой набор ключей и их значений, и который:

- содержит ключи типа string и значения заданного в шаблоне типа данных,
- увеличивает свой размер при добавлении в него элементов,
- предоставляет типовые свойства (количество ключей и значений, признак сортированного «словаря»),
- предоставляет типовые процедуры (вставка ключа и значения, удаление значения по ключу, поиск значения по ключу, очистка, сортировка),
- предоставляет типовые события, происходящие при работе со «словарем» (вставка, удаление, очистка).

Продемонстрировать использование класса.

Задача 16.

Разработать класс «XML-дерево» — TXmlTree — для представления XML-файла в памяти. Класс:

- содержит список тегов и значений типа string, а также список вложенных деревьев типа TXmlTree,
- обеспечивает загрузку и выгрузку XML-файла (без детализации),
- предоставляет типовые свойства (количество ключей и значений, количество вложенных деревьев),
- предоставляет типовые процедуры (вставка ключа и значения в заданную позицию, удаление значения по ключу, поиск значения по ключу или на основе «пути» по аналогии с файловой системой, очистка),
- предоставляет типовые события, происходящие при работе с контейнером (вставка, удаление, очистка),

Продемонстрировать использование класса.

Задача 17.

Разработать абстракцию файла TAbstractFile и два производных класса TOSFile (файл операционной системы) и TMemoryFile (файл в памяти). Классы:

- предоставляют типовые свойства (размер файла, текущая позиция указателя чтения-записи в файле),
- предоставляют типовые процедуры работы с файлом (запись, чтение, перемещение позиции указателя чтения-записи в файле, копирование содержимого другого файла),
- предоставляет типовые события, происходящие при работе с файлом (чтение, запись).

Продемонстрировать использование классов так: программно создать файл на диске с любым содержимым, скопировать содержимое файла на диске в файл в памяти.

Задача 18.

Разработать контейнер «битовый массив» — TBitArray, который

- содержит элементы типа Boolean так, что на каждый отводится лишь 1 бит,
- увеличивает свой размер при добавлении в него элементов,
- предоставляет типовые свойства (текущее количество элементов, количество зарезервированных элементов),
- предоставляет типовые процедуры, применяемые при работе с массивом (чтение и запись бита по индексу, расширение массива),
- при обращении к биту в массиве проверяет, что индекс находится в допустимых пределах.

Продемонстрировать использование класса.

Задача 19.

Разработать класс для представления матрицы произвольной размерности – TMatrix. Класс:

- содержит элементы типа double,
- принимает размеры матрицы на вход конструктора,
- предоставляет типовые свойства (размеры матрицы, элемент на пересечении строки и столбца),
- предоставляет типовые процедуры, применяемые при работе с матрицами (умножение, транспонирование),

Продемонстрировать использование класса.

Задача 20.

Разработать класс «кольцевой буфер» - TCircularBuffer. Класс:

- содержит байты данных,
- предоставляет типовые свойства (указатель чтения, указатель записи, количество данных в буфере),
- предоставляет типовые процедуры, применяемые при работе с кольцевым буфером (запись, чтение, наращивание размера буфера),
- генерирует исключительную ситуацию при переполнении буфера.

Продемонстрировать использование класса.

3.4. Представления оценочного средства в фонде

3.4.1. Вопросы для устного опроса (собеседование)

Наименование раздела: Модуль 1

1. Процедурное и объектно-ориентированное программирование.

2. Инкапсуляция.
3. Работа со ссылочными переменными. Сборка мусора.
4. Примитивные типы данных.
5. Оболочечные классы.

Наименование раздела: Модуль 2

1. Управляющие конструкции.
2. Функции. Модификаторы.
3. Локальные и глобальные переменные.
4. Переопределение методов.
5. Статическое и динамическое связывание.
6. Конструкторы. Базовый класс Object.
7. Массивы.
8. Исключительные ситуации.

3.4.2. Пример ситуационной задачи (или задачи)

Задание:

Разработать класс для представления матрицы произвольной размерности – TMatrix. Класс:

- содержит элементы типа double,
- принимает размеры матрицы на вход конструктора,
- предоставляет типовые свойства (размеры матрицы, элемент на пересечении строки и столбца),
- предоставляет типовые процедуры, применяемые при работе с матрицами (умножение, транспонирование),

Продемонстрировать использование класса.

3.5. Критериев оценивания контрольных заданий для использования в ФОС дисциплины

3.5.1. Критерии оценивания тестового задания:

Тестовые задания оцениваются по шкале: 1 балл за правильный ответ, 0 баллов за неправильный ответ. Итоговая оценка по тесту формируется путем суммирования набранных баллов и отнесения их к общему количеству вопросов в задании. Помножив полученное значение на 100%, можно привести итоговую оценку к традиционной следующим образом:

Процент правильных ответов Оценка

90 – 100% *От 9 до 10 баллов и/или «отлично»*

70 – 89 % *От 6 до 8 баллов и/или «хорошо»*

50 – 69 % *От 3 до 5 баллов и/или «удовлетворительно»*

менее 50 % *От 0 до 2 баллов и/или «неудовлетворительно»*

3.5.2. Критерии оценивания реферата (доклада):

От 4 до 5 баллов и/или «отлично»: глубокое и хорошо аргументированное обоснование темы; четкая формулировка и понимание изучаемой проблемы; широкое и правильное использование относящейся к теме литературы и примененных аналитических методов; содержание исследования и ход защиты указывают на наличие навыков работы студента в данной области; оформление работы хорошее с наличием расширенной библиографии; защита реферата (или выступление с докладом) показала высокий уровень профессиональной подготовленности студента;

От 2 до 3 баллов и/или «хорошо»: аргументированное обоснование темы; четкая формулировка и понимание изучаемой проблемы; использование ограниченного, но достаточного для проведения исследования количества источников; работа основана на среднем по глубине анализе изучаемой проблемы и при этом сделано незначительное

число обобщений; содержание исследования и ход защиты (или выступление с докладом) указывают на наличие практических навыков работы студента в данной области; реферат (или доклад) хорошо оформлен с наличием необходимой библиографии; ход защиты реферата (или выступления с докладом) показал достаточную профессиональную подготовку студента;

От 1 до 2 баллов и/или «удовлетворительно»: достаточное обоснование выбранной темы, но отсутствует глубокое понимание рассматриваемой проблемы; в библиографии преобладают ссылки на стандартные литературные источники; труды, необходимые для всестороннего изучения проблемы, использованы в ограниченном объеме; заметна нехватка компетентности студента в данной области знаний; оформление реферата (или доклада) содержит небрежности; защита реферата (или выступление с докладом) показала удовлетворительную профессиональную подготовку студента;

0 баллов и/или «неудовлетворительно»: тема реферата (или доклада) представлена в общем виде; ограниченное число использованных литературных источников; шаблонное изложение материала; суждения по исследуемой проблеме не всегда компетентны; неточности и неверные выводы по рассматриваемой литературе; оформление реферата (или доклада) с элементами заметных отступлений от общих требований; во время защиты (или выступления с докладом) студентом проявлена ограниченная профессиональная эрудиция.

3.5.3. Критерии оценивания на ситуационную задачу:

От 9 до 10 баллов и/или «отлично»: студент глубоко и полно владеет методами решения задачи; решение выполнено оптимальным способом; полученное решение соответствует условиям задачи; решение ситуационной задачи носит самостоятельный характер.

От 6 до 8 баллов и/или «хорошо»: решение студента соответствует указанным выше критериям, но в ход решения имеет отдельные неточности (несущественные ошибки); однако допущенные при решении ошибки исправляются самим студентом после дополнительных вопросов.

От 3 до 5 баллов и/или «удовлетворительно»: студент обнаруживает отсутствие навыков и понимание основных методик решения ситуационной задачи, но решение является неполным, имеет неточности и существенные ошибки; допущенные при решении ошибки не исправляются самим студентом после дополнительных вопросов.

От 0 до 2 баллов и/или «неудовлетворительно»: студент имеет разрозненные, бессистемные знания в области решаемой задачи; не владеет методами и подходами для решения задачи.

3.5.4. Критерии оценивания «Устный опрос»

- оценка «зачтено» выставляется студенту, если обладает систематизированными знаниями, умениями и навыками по данному разделу дисциплины;

- оценка «не зачтено» выставляется студенту, если он не проявил систематизированных знаний, умений и навыков по данному разделу дисциплины.

3.5.5. Критерий оценивания на зачет

Оценка «зачтено» ставится в том случае, если итоговый рейтинг студента составил 60 и более.

Оценка «не зачтено» ставится в том случае, если итоговый рейтинг студента составил менее 60 баллов.

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

Процедура оценки знаний умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, производится преподавателем в форме текущего контроля и промежуточной аттестации.

Для повышения эффективности текущего контроля и последующей промежуточной аттестации студентов осуществляется структурирование дисциплины на модули. Каждый модуль учебной дисциплины включает в себя изучение законченного раздела, части дисциплины.

Основными видами текущего контроля знаний, умений и навыков в течение каждого модуля учебной дисциплины являются устный опрос, подготовка рефератов, решение ситуационных задач, тестирование.

Студент должен выполнить все контрольные мероприятия, предусмотренные в модуле учебной дисциплины к указанному сроку, после чего преподаватель проставляет балльные оценки, набранные студентом по результатам текущего контроля модуля учебной дисциплины.

Контрольное мероприятие считается выполненным, если за него студент получил оценку в баллах, не ниже минимальной оценки, установленной программой дисциплины по данному мероприятию.

Промежуточная аттестация обучающихся проводится в форме зачета.

Зачет проводится для оценки уровня усвоения обучающимся учебного материала лекционных курсов и лабораторно-практических занятий, а также самостоятельной работы. Оценка выставляется или по результатам учебной работы студента в течение семестра, или по итогам письменного-устного опроса, или тестирования на последнем занятии. Для дисциплин и видов учебной работы студента, по которым формой итогового отчета является зачет, определена оценка «зачтено», «не зачтено».

Оценка «зачтено» ставится в том случае, если обучающийся:

- владеет знаниями, выделенными в качестве требований к знаниям обучающихся в области изучаемой дисциплины;
- демонстрирует глубину понимания учебного материала с логическим и аргументированным его изложением;
- владеет основным понятийно-категориальным аппаратом по дисциплине;
- демонстрирует практические умения и навыки в области исследовательской деятельности.

Оценка «не зачтено» ставится в том случае, если обучающийся:

- демонстрирует знания по изучаемой дисциплине, но отсутствует глубокое понимание сущности учебного материала;
- допускает ошибки в изложении фактических данных по существу материала, представляется неполный их объем;
- демонстрирует недостаточную системность знаний;
- проявляет слабое знание понятийно-категориального аппарата по дисциплине;
- проявляет непрочность практических умений и навыков в области исследовательской деятельности.

В этом случае студент сдаёт зачёт в форме устных и письменных ответов на любые вопросы в пределах освоенной дисциплины.

Основным методом оценки знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций является балльно-рейтинговая система, которая регламентируется положением «О балльно-рейтинговой системе оценки качества освоения образовательных программ в ФГБОУ ВО Белгородский ГАУ».

Основными видами поэтапного контроля результатов обучения студентов являются: входной контроль, текущий контроль, рубежный (промежуточный) контроль, творческий контроль, выходной контроль (зачет).

Уровень развития компетенций оценивается с помощью рейтинговых баллов.

Рейтинги	Характеристика рейтингов	Максимум баллов
Входной	Отражает степень подготовленности студента к изучению дисциплины. Определяется по итогам	5

	входного контроля знаний на первом практическом занятии.	
Рубежный	Отражает работу студента на протяжении всего периода изучения дисциплины. Определяется суммой баллов, которые студент получит по результатам изучения каждого модуля.	60
Творческий	Результат выполнения студентом индивидуального творческого задания различных уровней сложности, в том числе, участие в различных конференциях и конкурсах на протяжении всего курса изучения дисциплины.	5
Выходной	Является результатом аттестации на окончательном этапе изучения дисциплины по итогам сдачи экзамена. Отражает уровень освоения информационно-теоретического компонента в целом и основ практической деятельности в частности.	30
Общий рейтинг	Определяется путём суммирования всех рейтингов	100

Общий рейтинг по дисциплине складывается из входного, рубежного, выходного (зачета) и творческого рейтинга.

Входной (стартовый) рейтинг – результат входного контроля, проводимого с целью проверки исходного уровня подготовленности студента и оценки его соответствия предъявляемым требованиям для изучения данной дисциплины.

Он проводится на первом занятии при переходе к изучению дисциплины (курса, раздела). Оптимальные формы и методы входного контроля: тестирование, программированный опрос, в т.ч. с применением ПЭВМ и ТСО, решение комплексных и расчетно-графических задач и др.

Рубежный рейтинг – результат рубежного (промежуточного) контроля по каждому модулю дисциплины, проводимого с целью оценки уровня знаний, умений и навыков студента по результатам изучения модуля. Оптимальные формы и методы рубежного контроля: устные собеседования, письменные контрольные опросы, в т.ч. с использованием ПЭВМ и ТСО, результаты выполнения лабораторных и практических заданий. В качестве практических заданий могут выступать крупные части (этапы) курсовой работы или проекта, расчетно-графические задания, микропроекты и т.п.

Выходной рейтинг – результат аттестации на окончательном этапе изучения дисциплины по итогам сдачи зачета, проводимого с целью проверки освоения информационно-теоретического компонента в целом и основ практической деятельности в частности. Оптимальные формы и методы выходного контроля: письменные экзаменационные или контрольные работы, индивидуальные собеседования.

Творческий рейтинг – составная часть общего рейтинга дисциплины, представляет собой результат выполнения студентом индивидуального творческого задания различных уровней сложности.

В рамках рейтинговой системы контроля успеваемости студентов, семестровая составляющая балльной оценки по дисциплине формируется при наборе заданной в программе дисциплины суммы баллов, получаемых студентом при текущем контроле в процессе освоения модулей учебной дисциплины в течение семестра.

Итоговая оценка /зачёта/ компетенций студента осуществляется путём автоматического перевода баллов общего рейтинга в стандартные оценки.

Максимальная сумма рейтинговых баллов по учебной дисциплине составляет 100 баллов.

Оценка «зачтено» ставится в том случае, если итоговый рейтинг студента составил 60 и более.

Оценка «не зачтено» ставится в том случае, если итоговый рейтинг студента составил менее 60 баллов.

Не зачтено	Зачтено
менее 60 балла	60-100 баллов