



Федеральное государственное бюджетное образовательное учреждение высшего образования
«ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ ДВАЖДЫ ГЕРОЯ
СОВЕТСКОГО СОЮЗА, ЛЕТЧИКА-КОСМОНАВТА А.А. ЛЕОНОВА»

«УТВЕРЖДАЮ»

и.о. проректора

А.В. Троицкий

«__» _____ 2023 г.

***ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И
ТЕХНОЛОГИЙ***

***КАФЕДРА ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ***

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

**«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ
ПРОГРАММИРОВАНИЕ»**

Направление подготовки: 09.03.03 Прикладная информатика

Профиль: Прикладная информатика в системах управления

Уровень высшего образования: бакалавриат

Форма обучения: очная

Королев 2023

Рабочая программа является составной частью основной профессиональной образовательной программы и проходит рецензирование со стороны работодателей в составе основной профессиональной образовательной программы. Рабочая программа актуализируется и корректируется ежегодно.

Автор: Исаева Г.Н. Рабочая программа дисциплины: Объектно-ориентированное программирование – Королев МО: ТУ, 2023 г.

Рецензент: к.т.н. доцент Логачёва Н.В.

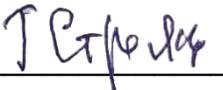
Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего профессионального образования (ФГО ВО) по направлению подготовки бакалавров 09.03.03 Прикладная информатика Учебного плана, утвержденного Ученым советом ТУ.

Протокол № 9 от 11.04.2023 г.

Рабочая программа рассмотрена и одобрена на заседании кафедры:

Заведующий кафедрой (ФИО, ученая степень, звание, подпись)	Артюшенко В.М., д.т.н. профессор 			
Год утверждения (переутверждения)	2023			
Номер и дата протокола заседания кафедры	№12 от 05.04.2023			

Рабочая программа согласована:

Руководитель ОПОП  Г.А. Стрельцова, к.т.н., доц.

Рабочая программа рекомендована на заседании УМС:

Год утверждения (переутверждения)	2023			
Номер и дата протокола заседания УМС	№ 5 от 11 апреля 2023 г.			

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОПОП

Целью изучения дисциплины является:

- развитие у студентов личностных качеств;
- формирование компетенций в соответствии с действующим образовательным стандартом по направлению подготовки 09.03.03 «Прикладная информатика».

В процессе обучения по данной дисциплине студент приобретает и совершенствует следующие компетенции:

Профессиональные компетенции:

- способен разрабатывать и адаптировать прикладное программное обеспечение (ПК-2);
- способен реализовывать проектные решения по созданию ИС на основе командных принципов работы (ПК-6).

Основными задачами дисциплины являются:

Подготовить выпускников к профессиональной **организационно-управленческой деятельности:**

- управленческая деятельность в подразделениях организаций, осуществляющих проектирование ИС;
- организация работы по взаимодействию с заказчиком ИС.

Подготовить выпускников к профессиональной **проектной деятельности:**

- проектирование ИС по видам обеспечения.

Содержание дисциплины включает освоение базовых методов разработки объектно-ориентированных программ.

Показатели освоения компетенций отражают следующие индикаторы:

Трудовые действия:

- Использует методы структурного и объектно-ориентированного программирования для разработки структуры программного кода ИС, верификации структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС;

- Использует CASE-средства и информационные технологии для управления командной работой, планирования деятельности, распределения поручений и контроля их исполнения.

Необходимые умения:

- Применяет языки программирования и работы с базами данных, инструменты и методы верификации структуры программного кода для разработки прототипа ИС в соответствии с требованиями;
- Применяет методики управления ИТ-проектами и современные инструменты и методы управления ИТ-проектами.

Необходимые знания:

- Анализирует языки программирования и работы с базами данных, инструменты и методы верификации структуры программного кода, современные объектно-ориентированные языки программирования, современные структурные языки программирования, языки современных бизнес-приложений для выполнения работ и управление работами по созданию (модификации) и сопровождению ИС;
- Понимает методы формирования команды, принципы организации командной работы.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Объектно-ориентированное программирование» относится к дисциплинам части, формируемой участниками образовательных отношений основной профессиональной образовательной программы подготовки бакалавров по направлению 09.03.03 «Прикладная информатика» и входит в состав модуля «Технологии программирования».

Изучение дисциплины базируется на знаниях, полученных в процессе изучения дисциплин «Линейная алгебра и аналитическая геометрия», «Теоретические основы информатики», «Языки программирования», «Введение в программную инженерию», «Основы алгоритмизации и программирования» и компетенциях: УК-1, УК-2, ОПК-1, УК-6, ПК-7, ПК-2, ПК-6, ОПК-2; ОПК-3; ОПК-7.

Знания и компетенции, полученные при освоении дисциплин модуля, являются базовыми при изучении дисциплин «Программные средства реализации информационных систем и информационных технологий», «Интернет-технологии», «Нейронные сети и нейрокомпьютеры» и при выполнении выпускной квалификационной работы бакалавра.

3. Объем дисциплины (модуля) и виды учебной работы

Общая трудоемкость дисциплины составляет 4 зачетные единицы, 144 часов. Преподавание дисциплины ведется в 4-м семестре и предусматривает проведение учебных занятий следующих видов: лекции, практические занятия, самостоятельная работа обучающихся, групповые и индивидуальные консультации.

Программой предусмотрены следующие виды контроля: два текущих контроля знаний в форме тестирования и промежуточную аттестацию в форме зачета с оценкой.

Таблица 1

Виды занятий	Всего часов	Семестр 4	Семестр 5	Семестр ...	Семестр ...
Общая трудоемкость	144	144			
ОЧНАЯ ФОРМА ОБУЧЕНИЯ					
Аудиторные занятия	48	48			
Лекции (Л)	16	16			
Практические занятия (ПЗ)	32	32			
Лабораторные работы (ЛР)	-	-			
Самостоятельная работа	96	96			
Курсовые, расчетно-графические работы	-	-	-		
Контрольная работа, домашнее задание	+	+			
	-	-			
Текущий контроль знаний (7 - 8, 15 - 16 недели)	Тест	Тест			
Вид итогового контроля	Зачёт с оценкой	Зачёт с оценкой			

4. Содержание дисциплины

4.1. Темы дисциплины и виды занятий

Таблица 2

Наименование тем	Лекции, час.	Практические занятия, час	Занятия (в том числе) в интерактивной форме, час	Практическая подготовка, час	Код компетенций
Тема 1. Введение в курс «Объект-	2	2	-	-	ПК-2

но-ориентированное программирование (ООП)»					
Тема 2. Принципы и методы технологии программирования	2	8	4	4	ПК-2
Тема 3. Технология ООП	4	8	4	4	ПК-6
Тема 4. Языки объектно-ориентированного программирования	4	8	4	4	ПК-6
Тема 5. Интегрированная среда разработки приложений Lazarus (Delphi 7).	4	6	4	4	ПК-6
Итого:	16	32	16	16	

4.2. Содержание тем дисциплины

Тема 1. Введение в курс «Объектно-ориентированное программирование (ООП)»

Основные понятия. Программирование. Программный продукт. Технология программирования. Технологические инструкции.

Тема 2. Принципы и методы технологии программирования

Технологический подход. Разработка программного продукта. Технологии программирования: со слабой формализацией, классические, гибкие.

Тема 3. Технология ООП

Вычислительная модель ООП. Структура объекта. Данные. Методы. Основные принципы объектно-ориентированного программирования: инкапсуляция, наследование, полиморфизм.

Тема 4. Языки объектно-ориентированного программирования.

Лексика языка. Базовые типы данных. Структурированные типы данных. Выражения и операции. Структура программы. Организация ввода/ вывода данных, в том числе из внешних источников.

Тема 5. Интегрированная среда разработки приложений Lazarus (Delphi 7).

Интерфейс среды. Разработка приложений в среде. Понятие проекта и его файловый состав. Средства управления параметрами проекта и среды разработки. Классы и библиотека среды. Архитектура приложений, имеющих доступ к внешним источникам данных.

5. Перечень учебно-методического обеспечения для самостоятельной работы по дисциплине

1. Методические указания для обучающихся по освоению дисциплины.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Структура фонда оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Объектно-ориентированное программирование» приведена в Приложении 1.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Рик, Г. Объектно-Ориентированное Программирование / Г. Рик ; под редакцией Н. Комлева. — Москва : СОЛОН-Пресс, 2023. — 298 с
2. Григорьев А.А. Методы и алгоритмы обработки данных : учеб. Пособие — М.: ИНФРА-М, 2021. — 383 с. / ЭБС «Знаниум». Режим доступа: URL <https://znanium.com/read?id=361208>

Дополнительная литература:

1. Хорев П.Б. Объектно-ориентированное программирование с примерами на C#. М.: Форум. 2022. – 200 с. ЭБС «Знаниум». Режим доступа: URL <https://znanium.com/read?id=351782>
2. Ачкасов, В. Программирование на Lazarus / В. Ачкасов. - 2-е изд., исправ. - Москва: Национальный Открытый Университет «ИНТУИТ», 2022. - 521 с. Режим доступа: URL: <http://biblioclub.ru/index.php?page=book&id=429187>

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Интернет-ресурсы:

<http://www.biblioclub.ru>

<http://znanium.com>

9. Методические указания для обучающихся по освоению дисциплины

Методические указания для обучающихся по освоению дисциплины, приведены в Приложении 2.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения:

Любая интегрированная среда программирования, поддерживающая язык Object Pascal, например, Lazarus (Delphi 7), MS Office 15.

Информационные справочные системы:

не предусмотрено курсом данной дисциплины.

Ресурсы информационно-образовательной среды ТУ.

Рабочая программа и методическое обеспечение по дисциплине «Объектно-ориентированное программирование».

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Лекционные занятия:

–аудитория, оснащенная презентационной техникой (проектор, экран), интерактивной доской Smart Board.

Практические занятия:

–учебный класс, оснащенный вычислительной техникой (ПК), современные лицензионные программно-технические средства: операционная система не ниже Windows XP; офисные программы MS Office 10; интегрированная среда разработки приложений Lazarus (Delphi 7), рабочее место преподавателя, оснащенное компьютером с доступом в Интернет;

–рабочие места студентов, оснащенные компьютерами с доступом в Интернет.

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ

*КАФЕДРА ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ*

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ
АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

(Приложение 1 к рабочей программе)

Направление подготовки: 09.03.03 Прикладная информатика

Профиль: Прикладная информатика в системах управления

Уровень высшего образования: бакалавриат

Форма обучения: очная

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

№ п./п.	Индекс компетенции	Содержание компетенции (или ее части)	Раздел дисциплины, обеспечивающий формирование компетенции (или ее части)	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции (или ее части), обучающийся должен:		
				трудовые действия	необходимые умения	необходимые знания
1.	ПК-2	способен разрабатывать и адаптировать прикладное программное обеспечение	Темы 1- 2	Использует методы структурного и объектно-ориентированного программирования для разработки структуры программного кода ИС, верификации структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС	Применяет языки программирования и работы с базами данных, инструменты и методы верификации структуры программного кода для разработки прототипа ИС в соответствии с требованиями	Анализирует языки программирования и работы с базами данных, инструменты и методы верификации структуры программного кода, современные объектно-ориентированные языки программирования, современные структурные языки программирования, языки современных бизнес-приложений для выполнения работ и управление работами по созданию (модификации) и сопровождению ИС
2.	ПК-6	способен реализовывать проектные решения по созданию ИС на основе командных	Темы 3- 5	Использует CASE-средства и информационные технологии для управления командной работой, планирования деятельности,	Применяет методики управления ИТ-проектами и современные инструменты и методы управления ИТ-проектами	Понимает методы формирования команды, принципы организации командной работы

		принципов работы		распределения поручений и контроля их исполнения		
--	--	---------------------	--	---	--	--

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Код компетенции	Инструменты, оценивающие сформированность компетенции	Показатель оценивания компетенции	Критерии оценки
ПК-2	Доклад в форме презентации	<p>А) полностью сформирована (компетенция освоена на высоком уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> •компетенция освоена на продвинутом уровне – 4 балла; •компетенция освоена на базовом уровне – 3 балла; <p>В) не сформирована (компетенция не освоена) – 2 и менее баллов</p>	<p>Проводится устно с использованием мультимедийных систем, а также с использованием технических средств</p> <p>Время, отведенное на процедуру – 10 - 15 мин.</p> <p>Неявка – 0.</p> <p>Критерии оценки:</p> <ol style="list-style-type: none"> 1.Соответствие представленной презентации заявленной тематике (1 балл). 2.Качество источников и их количество при подготовке доклада и разработке презентации (1 балл). 3.Владение информацией и способность отвечать на вопросы аудитории (1 балл). 4.Качество самой представленной презентации (1 балл). 5.Оригинальность подхода и всестороннее раскрытие выбранной тематики (1 балл). <p>Максимальная сумма баллов - 5 баллов.</p> <p>Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал.</p>
ПК-6	Расчетно-графические работы	<p>А) полностью сформирована (компетенция освоена на высоком уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> •компетенция освоена на продвинутом уровне – 4 балла; 	<p>Проводится на ПК в соответствующей среде разработки с составлением отчёта по работе.</p> <p>Критерии оценки:</p> <ol style="list-style-type: none"> 1. Оформление в соответствии с требованиями (1 балл); 2. Соответствует методическим указаниям в части

		<ul style="list-style-type: none"> •компетенция освоена на базовом уровне – 3 балла; В) не сформирована (компетенция не освоена) – 2 и менее баллов 	<p>структуры (1 балл);</p> <p>3. Содержание соответствует заявленной тематике (1 балл);</p> <p>4. Поставленные цели и задачи достигнуты (1 балл);</p> <p>5. Качественный и количественный состав использованных источников (1 балл). Максимальная оценка – 5 баллов</p> <p>Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал за каждую работу.</p>
ПК-2, ПК-6	Письменное задание	<p>А) полностью сформирована (компетенция освоена на высоком уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> •компетенция освоена на продвинутом уровне – 4 балла; •компетенция освоена на базовом уровне – 3 балла; <p>В) не сформирована (компетенция не освоена) – 2 и менее баллов</p>	<p>Проводится на ПК в соответствующей среде разработки с составлением отчёта по работе.</p> <p>Критерии оценки:</p> <ol style="list-style-type: none"> 1. Оформление в соответствии с требованиями (1 балл); 2. Соответствует методическим указаниям в части структуры (1 балл); 3. Содержание соответствует заявленной тематике (1 балл); 4. Поставленные цели и задачи достигнуты (1 балл); 5. Качественный и количественный состав использованных источников (1 балл). Максимальная оценка – 5 баллов <p>Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал за каждую работу.</p>
ПК-2, ПК-6	Контрольная работа (прово-	А) полностью сформирована (компетен-	Критерии оценки контрольной работы:

	<p>дится в качестве закрепления освоенного курса и компетенций) для студентов всех форм обучения</p>	<p>ция освоена на высоком уровне) – 5 баллов Б) частично сформирована: •компетенция освоена на продвинутом уровне – 4 балла; •компетенция освоена на базовом уровне – 3 балла; В) не сформирована (компетенция не освоена) – 2 и менее баллов</p>	<p>1.Соответствие содержания контрольной работы заявленной тематике (1 балл). 2.Качество источников и их количество при подготовке работы (1 балл). 3.Владение информацией и способность отвечать на вопросы аудитории (1 балл). 4.Качество самой представленной работы (1 балл). 5.Оригинальность подхода и всестороннее раскрытие выбранной тематики (1 балл). Максимальная сумма баллов - 5 баллов Оценка проставляется в журнал</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

Примерная тематика докладов в презентационной форме:

1. Классификация технологий программирования.
2. Современные парадигмы программирования.
3. Кризис в программировании в конце XX века.
4. Технология СОМ.
5. Первые языки ООП – обзор.
6. Современные языки программирования, поддерживающие объектно-ориентированный подход в разработке приложений.
7. Java – объектно-ориентированный ЯП ВУ.
8. История создания языка программирования ВУ С++.
9. Преимущество ООП.
10. Недостатки технологии ООП.
11. Современные интегрированные программные среды, поддерживающие технологию ООП.
12. Современные экономические задачи, решаемые с помощью методологии ООП.
13. Этапы проектирования сложных приложений в визуальных средах разработки.
14. Наследование, как инструмент создания иерархии классов.
15. Инкапсуляция и обеспечение безопасности разрабатываемого кода при ООП.
16. Перспективы развития технологии ООП.
17. Алгоритм Прима и его решение методом ООП.
18. Использование методологии ООП при решении функции «Аккермана».
19. Задача «Восемь ферзей» и её реализация в объектно-ориентированной среде программирования.
20. Реализация в объектно-ориентированной среде программирования рекурсивных графических алгоритмов.
21. Решение задачи «Кривая Коха» в объектно-ориентированной среде программирования.
22. Решение задачи «Кривая Кантора» в объектно-ориентированной среде программирования.
23. Решение задачи «Кривая Пеано» в объектно-ориентированной среде программирования.
24. Решение задачи «Треугольник Паскаля» в объектно-ориентированной среде программирования.
25. Рекурсивный алгоритм для головоломки «Ханойская башня» и его реализация на объектно-ориентированном ЯП.

Примерная тематика расчетно-графических работ:

1. Изменение параметров формы и создание простейших приложений в интегрированной среде разработки.
2. Средства управления параметрами проекта и среды разработки.
3. Характеристики встроенной системы отладки среды разработки.
4. Создание приложения «Блокнот» на основе компонента «Мето».
5. Разработка калькулятора с добавлением таких функций, как определение числа на чётность, нахождение периметра и площади простейшей геометрической фигуры.
6. Этапы разработки приложения «Редактор текстов».
7. Модификация приложения «Редактор текста», обеспечивающая возможность изменения шрифта выделенного в окне редактирования текста.
8. Модификация приложения «Редактор текста», обеспечивающая добавление в меню сервисных функций: ликвидацию лишних пробелов между словами; ликвидацию повторяющихся знаков препинания (две точки подряд, две запятые подряд и т. п.).
9. Дополнение меню приложения «Редактор текста» функциями сбора статистики по отображенному в окне редактирования тексту (подсчету количества символов и слов).
10. Дополнение меню приложения «Редактор текста» функциями сбора статистики по отображенному в окне редактирования тексту (подсчету количества слов и предложений).
11. Дополнение меню приложения «Редактор текста» функциями сбора статистики по отображенному в окне редактирования тексту (подсчету количества предложений и абзацев).
12. Модификация приложения «Редактор текста», обеспечивающая добавление в меню сервисных функций: проверку правильности написания первого слова в предложении (обязательно с прописной буквы); преобразование всех букв выделенного текста в прописные (строчные).
13. Разработка приложений с использованием компонентов RadioGroup и Panel (Панель).
14. Создание приложения «Простейший плеер» в интегрированной среде разработки приложений.
15. Использование графических файлов в интегрированной среде разработки приложений.
16. Построение гистограмм с использованием внешних источников хранения данных.
17. Описание архитектуры приложений, работающих с внешними источниками данных (базами данных).

18. Создание приложения, работающего с книгой MS Excel.
19. Модификация приложения — пример работы с книгой MS Excel: дополнение его функцией подсчета нагрузки (количества часов занятий) для заданного преподавателя (всего и отдельно по видам занятий).

Используя технологию ООП разработать приложения, которые содержат следующие сервисные компоненты в интерфейсной части:

20. Всплывающее меню.
21. Модальные окна
22. Двигающиеся объекты
23. Удобные окна для работы с массивами различных типов.
24. Списки выбора и выпадающие списки
25. Диалоги

Примерная тематика письменного задания:

Используя технологию ООП разработать приложения, которые содержат следующие сервисные компоненты в интерфейсной части:

Варианты (по списку журнала)	Компоненты
1-3	Всплывающее меню
4-6	Графики и рисунки, полученные из внешних источников
7-10	Модальные окна
11-14	Двигающиеся (динамические) объекты
15-18	Удобные окна для работы с массивами различных типов.
19-21	Таблицы, импортированные из БД SQL Server
22-23	Списки выбора и выпадающие списки
24-25	Диалоги
Во всех вариантах предусмотреть удобное обращение для пользователя к данным и информации, включить в приложение кнопки, метки, ссылки	

Тематика контрольной работы приведена в Приложении 2

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

Формой контроля знаний по дисциплине «Объектно-ориентированное программирование» являются две текущие аттестации в виде тестов и одна промежуточная в виде зачёта с оценкой.

Неделя текущего контроля	Вид оценочного средства	Код компетенций, оценивающий знания, умения, навыки	Содержание оценочного средства	Требования к выполнению	Срок сдачи (неделя семестра)	Критерии оценки по содержанию и качеству с указанием баллов
Согласно графику учебного процесса	тестирование	ПК-2, ПК-6.	25 вопросов	Компьютерное тестирование; время, отведенное на процедуру -30 минут,	Результаты тестирования предоставляются в день проведения процедуры	Критерии оценки определяются процентным соотношением. Не явка - Удовлетворительно - от 51% правильных ответов. Хорошо - от 75%. Отлично – от 85%.
Согласно графику учебного процесса	Зачёт с оценкой	ПК-2, ПК-6	2 вопроса	Зачет проводится в устной форме, путем ответа на вопросы. Время, отведенное на процедуру – 20 минут.	Результаты предоставляются в день проведения зачета	Критерии оценки: «Отлично»: • знание основных понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответ на вопросы билета. «Хорошо»: • знание основных понятий предмета; • умение

						<p>использовать и применять полученные знания на практике;</p> <ul style="list-style-type: none"> • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответы на вопросы билета • неправильно решено практическое задание <p>«Удовлительно»:</p> <ul style="list-style-type: none"> • демонстрирует частичные знания по темам дисциплин; • незнание неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; <p>«Неудовлительно»:</p> <ul style="list-style-type: none"> • демонстрирует частичные знания по темам дисциплин; • незнание основных понятий предмета; • неумение использовать и приме-
--	--	--	--	--	--	--

						нять полученные знания на практике; • не работал на практических занятиях; • не отвечает на вопросы.
--	--	--	--	--	--	--

** Итоговое начисление баллов по дисциплине осуществляется в соответствии с разработанной и внедренной балльно-рейтинговой системой контроля и оценивания уровня знаний и внеучебной созидательной активности обучающихся, согласно приказу «О внедрении новой балльно-рейтинговой системы контроля и оценивания уровня знаний и внеучебной созидательной активности обучающихся».*

4.1. Типовые вопросы, выносимые на тестирование

Тесты используются в режиме промежуточного контроля. По форме заданий выбраны закрытые тесты (с выборочным ответом). Каждому вопросу соответствует один вариант ответа.

Создание приложений Windows в среде Lazarus (Delphi).

1. Исходный код модуля хранится в файле с расширением:

- (?) pas
- (?) res
- (?) exe
- (?) dpr

2. Выберите основные расширения файлов, создаваемых при разработке приложения в среде Delphi:

- (?) «. tpu»
- (?) «. dpr»
- (?) «. pas»
- (?) «. dfm»

3. Класс в Delphi может кроме разделов private и public иметь разделы:

- (?) protected
- (?) automated
- (?) tobject
- (?) published

4. Основная особенность экземпляров объектов в Delphi / Lazarus состоит в том, что они по умолчанию являются...

- (?) динамическими

- (?) статическими
- (?) смешанными
- (?) стандартными

5. При создании формы используется событие:

- (?) «OnClick»
- (?) «OnKeyUp»
- (?) «OnKeyPress»
- (?) «OnCreate»
- (?) «OnClose»

6. При необходимости отрисовки формы используется событие:

- (?) «OnClick»
- (?) «OnKeyUp»
- (?) «OnCreate»
- (?) «OnPaint»
- (?) «OnClose»

7. Какие действия осуществляются в результате выполнения следующего фрагмента программного кода? `type TNumber = class n: integer; constructor Create (n_ : integer); end;`

- (?) объявляется класс «целое число»
- (?) объявляется класс «вещественное число»
- (?) объявляется класс «форма»
- (?) объявляется класс «счётчик»

8. Если конструктор или деструктор в классе не задан, то...

- (?) конструктор наследуется от потомка, а деструктор - от предка
- (?) конструктор наследуется от предка, а деструктор - от потомка
- (?) они оба наследуются от предка

9. Если конструктор или деструктор в классе не задан, то они наследуются от...

- (?) TNumber
- (?) TObject
- (?) TList
- (?) Tmenu

10. В Delphi 7.0 или Lazarus панель, содержащая библиотеку компонентов, называется

- (?) палитрой
- (?) вкладкой
- (?) окном инспектора объектов
- (?) окном приложения

11. С помощью инспектора объектов в Delphi 7.0 (Lazarus) задаются ...

- (?) свойства компонентов и обработчики событий
- (?) свойства компонентов
- (?) обработчики событий
- (?) окна

12. Библиотека VCL представляет собой иерархию объектов, имеющую общего предка - ...

- (?) Events
- (?) TObject
- (?) Properties
- (?) DObject

Технология объектно-ориентированного программирования (ООП).

1. Какие этапы включает технология программирования?

- (?) указание последовательности выполнения технологических операций
- (?) перечисление условий, при которых выполняется та или иная операция
- (?) подробное описание самих операций
- (?) множество интерфейсов операций

2. Объект является экземпляром...

- (?) класса
- (?) переменной
- (?) объекта
- (?) метода

3. Что определяет любая технология в программировании?

- (?) способ описания модели, используемой на конкретном этапе разработки
- (?) свойства моделируемого объекта
- (?) способ связи совокупности данных
- (?) способ описания интерфейсов

4. Абстрагирование – это

- (?) выделение существенных с точки зрения рассмотрения свойств объекта
- (?) сокрытие деталей реализации объекта
- (?) совокупность данных
- (?) совокупность интерфейсов

5. Инкапсуляция - это

- (?) моделирование объекта
- (?) абстрагирование
- (?) сокрытие деталей реализации объекта
- (?) показ деталей реализации

6. Наследование в объектно-ориентированной среде позволяет создавать:

- (?) иерархии объектов
- (?) доступ к исходному коду родительского объекта
- (?) несколько предков для одного объекта
- (?) абстракции

7. Полиморфизм – это

- (?) объединение в единое целое данных и алгоритмов обработки этих данных
- (?) свойство объектов порождать своих потомков
- (?) выделение существенных свойств объекта
- (?) свойство родственных объектов решать схожие по смыслу проблемы разными способами

8. Укажите, какую операцию поддерживает «чистое» объектно-ориентированное программирование?

- (?) послать объекту O сообщение Mes с параметрами P1 ... Pn ;
- (?) построить иерархию объектов O1 ... On;
- (?) описать параметры P1 ... Pn объекта. O;

9. Основными принципами ООП являются:

- (?) Полиморфизм
- (?) Инкапсуляция
- (?) Наследование
- (?) Программирование без «GOTO»

10. Основными достоинствами объектно-ориентированного программирования по сравнению с модульным программированием является:

- (?) полная модульность программного кода
- (?) декомпозиция программного обеспечения, наиболее приближенная к объективной реальности
- (?) механизмы ООП позволяют конструировать сложные объекты из сравнительно простых
- (?) возможность практически независимой разработки отдельных частей (объектов) программы

11. Выберите среды, поддерживающие визуальное программирование:

- (?) Visual Studio.NET
- (?) Borland Pascal
- (?) Visual Basic
- (?) C++ Builder
- (?) Delphi

12. При какой технологии программирования показатель повторного использования программного кода выше?

- (?) при структурной
- (?) при модульной
- (?) при объектно-ориентированной
- (?) при всех существующих практически одинаков

13. Объект ООП – это

- (?) совокупность входных параметров
- (?) совокупность процедур, моделирующих его поведение
- (?) совокупность переменных состояния и связанных с ними методов (операций)
- (?) совокупность данных, характеризующих его состояние

4.2. Типовые вопросы, выносимые на зачёт

1. Принципы и методы технологии программирования.
2. Классификация технологий программирования.
3. Вычислительная модель ООП.
4. Современная технология программирования — компонентный подход.
5. Основные правила программирования в объектно-ориентированных средах.
6. Документирование программы.
7. Понятие класса и объекта.
8. Описание объекта. Поля и методы объекта.
9. Основные принципы ООП: инкапсуляция.
10. Основные принципы ООП: наследование.
11. Основные принципы ООП: полиморфизм.
12. Абстрактные классы.
13. Понятие вложенных классов и композиция.
14. Понятие агрегации и контейнера.
15. Конструкторы и деструкторы.
16. Событийно-управляемое программирование.
17. Среда разработки приложений Lazarus (Delphi) – общие сведения.
18. Интерфейс среды Lazarus (Delphi).
19. Основные возможности языка ObjectPascal.

20. Типы языка ObjectPascal
21. Объектная модель Lazarus (Delphi).
22. Файлы проекта.
23. Этапы создания приложений Windows в среде Lazarus (Delphi).
24. Модуль. Структура модуля.
25. Библиотека классов VCL.
26. События и свойства.
27. Элементы управления в среде разработки.
28. Диалоги в Lazarus (Delphi).
29. Работа с базами данных в Lazarus (Delphi).
30. Обработка исключений в Lazarus (Delphi).

*ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И
ТЕХНОЛОГИЙ
КАФЕДРА ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ*

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ. ДЛЯ ОБУЧАЮЩИХСЯ ПО
ОСВОЕНИЮ ДИСЦИПЛИНЫ
«ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»
(Приложение 2 к рабочей программе)**

Направление подготовки: 09.03.03 Прикладная информатика
Профиль: Прикладная информатика в системах управления
Уровень высшего образования: бакалавриат
Форма обучения: очная

Королев 2023

1. Общие положения

Целью изучения дисциплины является:

- развитие у студентов личностных качеств;
- формирование компетенций в соответствии с действующим образовательным стандартом по направлению подготовки 09.03.03 «Прикладная информатика».

Основными задачами дисциплины являются:

Подготовить выпускников к профессиональной **организационно-управленческой деятельности:**

- управленческая деятельность в подразделениях организаций, осуществляющих проектирование ИС;
- организация работы по взаимодействию с заказчиком ИС.

Подготовить выпускников к профессиональной **проектной деятельности:**

- проектирование ИС по видам обеспечения.

Содержание дисциплины включает освоение базовых методов разработки объектно-ориентированных программ.

2. Указания по проведению практических занятий

Тема: Введение в курс «Объектно-ориентированное программирование (ООП)»

Практическое занятие 1

Вид практического занятия: *подготовка доклада.*

Тема и содержание практического занятия:

Образовательные технологии: диспут.

Цель работы: Самостоятельная работа студентов по заранее выданным вопросам и обсуждение представленного материала на занятии в форме семинара.

Основные положения темы занятия:

1. Область применимости технологии ООП
2. Исторические предпосылки ООП

Вопросы для обсуждения:

1. Кризис программирования 20 века
2. Первые языки технологии ООП
3. Первые программные среды, поддерживающие ООП

Продолжительность занятия – **2 ч.**

Тема: Принципы и методы технологии программирования

Практическое занятие 2

Вид практического занятия: *подготовка доклада.*

Тема и содержание практического занятия:

Образовательные технологии: групповая дискуссия.

Цель работы: Самостоятельная работа студентов по заранее выданным вопросам и обсуждение представленного материала на занятии в форме семинара.

Основные положения темы занятия:

3. Современная классификация технологий программирования.
4. Принципы и методы технологии программирования.

Вопросы для обсуждения:

4. Программный продукт.
5. Программное обеспечение.
6. Технологический подход.
7. Этапы развития технологии программирования.
8. Процессы и стадии разработки современного программного обеспечения.

Продолжительность занятия – **8 ч**

Тема: Технология ООП

Практическое занятие 3

Вид практического занятия: *смешанная форма практического занятия.*

Тема и содержание практического занятия:

Образовательные технологии: дебаты (активное высказывание по позициям и обсуждение спорных вопросов).

Цель работы: Получение практических навыков и освоение основных принципов работы при написании объектно-ориентированного программного кода.

Основные положения темы занятия:

1. Статическая и динамическая часть структуры объекта
2. Реализация и преимущества основных принципов объектно-ориентированного программирования.

Вопросы для обсуждения:

1. Структура объекта.
2. Интерфейс объекта.
3. Скрытие деталей реализации объекта.
4. Создание иерархии объектов.
5. Полиморфизм.

Продолжительность занятия – 8 ч.

Тема: Языки объектно-ориентированного программирования.

Практическое занятие 4

Вид практического занятия: *смешанная форма практического занятия.*

Тема и содержание практического занятия:

Образовательные технологии: групповая дискуссия.

Цель работы: Получение практических навыков в постановке, решении, отладке, тестировании практических задач в среде программирования, поддерживающей Object Pascal.

Основные положения темы занятия:

1. Структура программы на Object Pascal и её основные разделы.
2. Примеры задач с различными алгоритмами, демонстрирующими возможности языка программирования.

Вопросы для обсуждения:

1. Простые и структурированные типы данных в Object Pascal.
2. Выражения, операции, операторы языка.
3. Подключение стандартных библиотек и модулей, разработанных пользователем.
4. Ввод/ вывод данных.
5. Работа с файлами.

Продолжительность занятия – 8 ч.

7). **Тема:** Интегрированная среда разработки приложений Lazarus (Delphi 7).

Практическое занятие 5

Вид практического занятия: *смешанная форма практического занятия.*

Тема и содержание практического занятия:

Образовательные технологии: групповая дискуссия.

Цель работы: Создание приложений различной степени сложности в интегрированной среде разработки приложений Lazarus (Delphi 7).

Основные положения темы занятия:

1. Понятие проекта.
2. Средства управления параметрами проекта.
3. Архитектура приложений, работающих с внешними источниками данных.

Вопросы для обсуждения:

1. Интерфейс среды.
2. События и свойства.
3. Инспектор дерева объектов.
4. Файлы проекта, программного кода, кода формы.
5. Создание классов.
6. Работа с диалогами и БД.

Продолжительность занятия – **6 ч.**

3. Указания по проведению лабораторного практикума

Не предусмотрен учебным планом.

4. Указания по проведению самостоятельной работы студентов

Цель самостоятельной работы: подготовить бакалавров к самостоятельному научному творчеству.

Задачи самостоятельной работы:

- расширить знания по современным языкам и средам программирования;
- овладеть технологией объектно-ориентированного программирования;
- систематизировать знания, подходы, методы по решению прикладных задач пользователя с помощью ВС.

Виды самостоятельной работы представлены в таблице 1.

Таблица 1

п/п	Наименование блока (раздела) дисциплины	Виды СРС
	Тема 1 Введение в курс «Объектно-ориентированное программирование (ООП)»	<p>Самостоятельное изучение тем, подготовка рефератов. Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Область применения ООП. 2. Обзор различных подходов к построению сред программирования, поддерживающих ООП. 3. История развития технологий программирования. <p>Примерная тематика рефератов:</p> <ol style="list-style-type: none"> 1. Современные постановки задач ООП 2. Основные парадигмы программирования. 3. Кризис программирования двадцатого века. 4. Истоки зарождения основ ООП.
	Тема 2. Принципы и методы технологии программирования	<p>Самостоятельное изучение тем, создание презентаций. Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Задачи, решаемые методом ООП. 2. Основные аспекты технологии программирования(ТП). 3. Элементы программной инженерии, как основа ТП. <p>Примерная тематика докладов с презентацией:</p> <ol style="list-style-type: none"> 1. Задачи, использующие классы библиотеки Math. 2. Задачи, использующие классы библиотеки работы с массивами. 3. Задачи, использующие классы библиотеки работы с текстовой информацией
	Тема 3. Технология ООП.	<p>Самостоятельное изучение тем, создание презентаций. Вопросы, выносимые на самостоятельное изучение:</p> <ol style="list-style-type: none"> 1. Рекурсивные алгоритмы 2. История развития рекурсивных алгоритмов. 3. Область применения рекурсивных алгоритмов, и их реализация методом ООП <p>Примерная тематика докладов для презентаций:</p> <ol style="list-style-type: none"> 1. Алгоритм Прима 2. Задача о восьми ферзях

		3. Кривая Коха 4. Кривая Пеано
	Тема 4. Языки объектно-ориентированного программирования.	Самостоятельное изучение тем, подготовка рефератов. Вопросы, выносимые на самостоятельное изучение: 1. Реализация классов, контейнеров, агрегатов. 2. Область применения Object Pascal для решения задач методом ООП. Примерная тематика рефератов: 1. Сравнение подходов ООП и структурного программирования. 2. Основные правила программирования на Object Pascal 3. Создание приложений на языке ООП в различных областях экономической и инженерно-технической деятельности.
	Тема 5. Интегрированная среда разработки приложений Lazarus (Delphi 7).	Самостоятельное изучение тем, создание презентаций, подготовка реферата. Вопросы, выносимые на самостоятельное изучение: 1. Задачи, решающие проблемы работы с внешними источниками данных 2. Задачи, обеспечивающие работу приложений с доступом к БД Примерная тематика рефератов 1. Доступ к файлам 2. Работа с диалогами 3. Подключение базы данных MS Access. 4. Подключение БД MySQL

5. 5. Указания по проведению контрольных работ для студентов

5.1. Требования к структуре

Структура контрольной работы должна способствовать раскрытию темы: иметь титульный лист, содержание, введение, основную часть, заключение, список литературы.

5.2. Требования к содержанию (основной части)

1. Во введении обосновывается актуальность темы, определяется цель работы, задачи и методы исследования.

2. При определении целей и задач исследования необходимо правильно их формулировать. Так, в качестве цели не следует употреблять глагол «сделать». Правильно будет использовать глаголы: «раскрыть», «определить», «установить», «показать», «выявить» и т.д.

3. Основная часть работы включает 2 - 4 вопроса, каждый из которых посвящается решению задач, сформулированных во введении, и заканчивается констатацией итогов.

4. Приветствуется иллюстрация содержания работы таблицами, графическим материалом (рисунками, схемами и т.п.).

5. Необходимо давать ссылки на используемую Вами литературу.

6. Заключение должно содержать сделанные автором работы выводы, итоги исследования.

7. Вслед за заключением идет список литературы, который должен быть составлен в соответствии с установленными требованиями. Если в работе имеются приложения, они оформляются на отдельных листах, и должны быть соответственно пронумерованы.

5.3. Требования к оформлению

Объём контрольной работы – 10-12 страниц формата А4, напечатанного с одной стороны текста (1,5 интервал, шрифт Times New Roman).

Тематика контрольных работ

Тема: Решение практических задач методом ООП в системе визуального объектно-ориентированного программирования Delphi	
В каждом варианте задания должно быть реализовано приложение Windows, заголовков главного окна которого должен содержать: ФИО, группу и номер варианта.	
1.	Создать меню с командами: Input, Calc, Exit При выборе команды Input открывается диалоговое окно, содержащее: <ul style="list-style-type: none">– три поля ввода типа TEdit;– группу из двух флажков («периметр», «площадь») типа TCheckBox;– кнопку типа TButton (свойство Default имеет значение true). Обеспечить возможность: <ul style="list-style-type: none">– ввода длин трех сторон треугольника;– выбора режима с помощью флажков: подсчет периметра и (или) площади треугольника. При выборе команды Calc открывается диалоговое окно с результатами.
2.	Создать меню с командами: Begin, Help, About . При выборе команды Begin открывается диалоговое окно, содержащее: <ul style="list-style-type: none">– поле ввода типа TEdit меткой Input;– метку типа TLabel для вывода результата;– группу из трех переключателей (2, 16, 8) типа TRadioGroup;– две кнопки типа TButton — Done OK. Обеспечить возможность: <ul style="list-style-type: none">– ввода числа в поле Input;– выбора режима преобразования с помощью переключателей: перевод в двоичную, шестнадцатеричную или восьмеричную

	<p>систему счисления. При щелчке на кнопке Do должен появляться результат перевода.</p>
3.	<p>Создать меню с командами: Inputcolor, Change, Exit, Help. При выборе команды Exit приложение завершает работу. При выборе команды Inputcolor открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – три поля ввода типа TEdit метками Red, Green, Blue; – группу из трех флажков (Left, Right) типа TCheckBox; – кнопку типа TButton. <p>Обеспечить возможность ввода RGB-составляющих цвета. При выборе команды Change, цвет фона главного окна изменяется на заданный (левая, правая или обе половины окна в зависимости от установки флажков).</p>
4.	<p>Создать меню с командами: Inputcolor, Change, Clear. При выборе команды Inputcolor открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – окно-список для выбора цвета типа TListBox; – группу из трех переключателей (Red, Green, Blue) типа TRadioButtonGroup; – кнопку типа TButton. <p>Обеспечить возможность:</p> <ul style="list-style-type: none"> – выбора цвета; – ввода режима, определяющего, какая область закрашивается: все окно, его верхняя или нижняя половина. <p>При выборе команды Change цвет фона главного окна изменяется на заданный (верхняя, нижняя или обе половины в зависимости от введенного режима). При выборе команды Clear восстанавливается первоначальный цвет фона окна.</p>
5.	<p>Написать приложение Windows, которое выполняет анимацию изображения. Создать меню с командами: Show picture, Choose, Animate, Stop, Quit. Команда Quit завершает приложение. При выборе команды Showpicture в центре экрана рисуется объект, состоящий из нескольких графических примитивов. При выборе команды Choose открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – поле ввода типа TEdit с меткой Speed для ввода скорости движения объекта; – группу Direction из двух переключателей (Up-Down, Left-Right) типа TRadioButtonGroup для выбора направления движения; – кнопку типа TButton. <p>По команде Animate объект начинает перемещаться в выбранном</p>

	<p>направлении до края окна и обратно с заданной скоростью, по команде Stop — прекращает движение.</p>
6.	<p>Написать приложение Windows, которое по заданным в файле исходным данным строит график или столбиковую диаграмму. Создать меню с командами: Input data, Choose, Line, Bar, Quit. Команды Line, Bar запрещены. Команда Quit завершает приложение. При выборе команды Input data из файла читаются исходные данные (файл сформировать самостоятельно).</p> <p>По команде Choose открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – окно-список для выбора цвета графика типа TListBox; – группу из двух переключателей (Line, Bar) типа TRadioGroup; – кнопку типа TButton. <p>Обеспечить возможность ввода цвета и выбора режима: построение графика (Line) или столбиковой диаграммы (Bar). После указания параметров разрешается соответствующая команда меню.</p> <p>По команде Line или Bar в главном окне приложения выбранным цветом строится график или диаграмма. Окно должно содержать заголовок графика или диаграммы, наименование и градуацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна</p>
7.	<p>Написать приложение Windows, которое по заданным в файле исходным данным выводит информацию о портативных компьютерах. Создать меню с командами: Notebooks, YourChoice, Quit. Команда YourChoice запрещена. Команда Quit завершает приложение. При запуске приложения из файла читаются исходные данные. Файл необходимо сформировать самостоятельно. Каждая строка файла содержит тип компьютера, цену (Price) и емкость жесткого диска (HardDrive). При выборе команды Show5 по \л/ открывается диалоговое окно, содержащее:</p> <p>окно типа TStringGrid с введенной информацией;</p> <p>заголовки граф типа TLabel;</p> <p>поле типа TEdit для ввода минимальной емкости диска;</p> <p>поле типа TEdit для ввода максимальной приемлемой цены;</p> <p>группу из двух переключателей (HardDrive, Price) типа TRadioGroup;</p> <p>ОК, Cancel — кнопки типа TBitBtn.</p> <p>После ввода всех данных команда меню YourChoice разрешается. Команда YourChoice открывает диалоговое окно, содержащее список компьютеров, удовлетворяющий введенным ограничениям и упорядоченный по отмеченной характеристике.</p>
8.	<p>Написать приложение Windows, — графическую иллюстрацию сортировки методом выбора.</p> <p>Создать меню с командами: File, Animate, About, Exit. Команда Animate запрещена. Команда Exit завершает приложение. Команда About открывает окно с информацией о разработчике. Для вы-</p>

	<p>бора файла исходных данных (команда File) использовать объект <code>OpenDialog</code>.</p> <p>Из выбранного файла читаются исходные данные для сортировки (сформировать самостоятельно не менее трех файлов различной длины с данными целого типа).</p> <p>После чтения данных разрешается команда меню Animate. При выборе команды Animate в главном окне приложения отображается процесс сортировки в виде столбиковой диаграммы. Каждый элемент представляется столбиком соответствующего размера. На каждом шаге алгоритма два элемента меняются местами. Окно должно содержать заголовков. Изображение должно занимать все окно.</p>
9.	<p>Написать приложение Windows, — графическую иллюстрацию аппроксимации методом наименьших квадратов зависимости</p> $y = a*x + b*x*x + c*\log_2x.$ <p>Создать меню с командами: Open, Coefficients, Show, About, Exit. Команда Exit завершает приложение. Команда About открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда Open) использовать объект <code>OpenDialog</code>. Исходные данные для аппроксимации — массивы экспериментальных значений аргумента x и функции $y(x)$ — сформировать самостоятельно.</p> <p>При выборе команды Coefficients выводится окно сообщений с вычисленными коэффициентами, a, b и c. При выборе команды Show в главном окне приложения отображается график зависимости и исходные данные в виде точек. Окно должно содержать заголовков. Изображение должно занимать все окно.</p>
10.	<p>Написать анализатор текстовых файлов, выводящий информацию о количестве слов в тексте, а также статистическую информацию о введенной пользователем букве.</p> <p>Создать меню:</p> <ul style="list-style-type: none"> – Файл <ol style="list-style-type: none"> 1. Загрузить текст 2. Выход 3. Анализ 4. Количество слов 5. Повторяемость буквы <p>При выборе файла для загрузки использовать объект типа <code>OpenDialog</code>. Для отображения текста использовать объект типа <code>Memo</code>. При выборе команды <i>Количество слов</i> программа должна вывести в окно сообщений количество слов в тексте. При выборе команды <i>Повторяемость буквы</i> программа предлагает пользователю. Ввести букву, а затем выводит количество ее повторений без учета регистра в окно сообщений.</p>
11.	<p>Написать приложение Windows, — графическую иллюстрацию сортировки методом «пузырька».</p> <p>Создать меню с командами: File, Animate, About, Exit.</p>

	<p>Команда Animate запрещена. Команда Exit завершает приложение. Команда About открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда File) использовать объект <code>OpenDialog</code>.</p> <p>Из выбранного файла читаются исходные данные для сортировки (сформировать самостоятельно не менее трех файлов различной длины с данными целого типа).</p> <p>После чтения данных разрешается команда меню Animate. При выборе команды Animate в главном окне приложения отображается процесс сортировки в виде столбиковой диаграммы. Каждый элемент представляется столбиком соответствующего размера. На каждом шаге алгоритма два элемента меняются местами. Окно должно содержать заголовок. Изображение должно занимать все окно.</p>
12.	<p>Написать приложение <code>Windows</code>, — графическую иллюстрацию сортировки методом вставки.</p> <p>Создать меню с командами: File, Animate, About, Exit.</p> <p>Команда Animate запрещена. Команда Exit завершает приложение. Команда About открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда File) использовать объект <code>OpenDialog</code>.</p> <p>Из выбранного файла читаются исходные данные для сортировки (сформировать самостоятельно не менее трех файлов различной длины с данными целого типа).</p> <p>После чтения данных разрешается команда меню Animate. При выборе команды Animate в главном окне приложения отображается процесс сортировки в виде столбиковой диаграммы. Каждый элемент представляется столбиком соответствующего размера. На каждом шаге алгоритма два элемента меняются местами. Окно должно содержать заголовок. Изображение должно занимать все окно.</p>
13.	<p>Создать меню с командами: Begin, Help, Inputcolor, About.</p> <p>При выборе команды Begin открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – поле ввода типа <code>TEdit</code> меткой Input; – метку типа <code>TLabel</code> для вывода результата; – группу из трех переключателей (2, 16, 8) типа <code>TRadioGroup</code>; – две кнопки типа <code>TButton</code> — <code>Do</code> и <code>OK</code>. <p>Обеспечить возможность:</p> <ul style="list-style-type: none"> – ввода числа в поле Input; – выбора режима преобразования с помощью переключателей: перевод в двоичную, шестнадцатеричную или восьмеричную систему счисления. <p>При выборе команды Inputcolor открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – окно-список для выбора цвета типа <code>TListBox</code>;

	<ul style="list-style-type: none"> – группу из трех переключателей (Red, Green, Blue) типа TRadioButtonGroup; – кнопку типа TButton. <p>Обеспечить возможность:</p> <ul style="list-style-type: none"> – выбора цвета; – ввода режима, определяющего, какая область закрашивается: все окно, его верхняя или нижняя половина.
14.	<p>Написать анализатор текстовых файлов, выводящий информацию о количестве слов в тексте.</p> <p>Создать меню: Файл</p> <ol style="list-style-type: none"> 1. Загрузить текст 2. Выход 3. Анализ 4. Количество слов <p>При выборе файла для загрузки использовать объект типа OpenFileDialog. Для отображения текста использовать объект типа Memo. При выборе команды <i>Количество слов</i> программа должна вывести в окно сообщений количество слов в тексте.</p>
15.	<p>Создать меню с командами: Inputcolor, Color, Exit, Help.</p> <p>При выборе команды Exit приложение завершает работу.</p> <p>При выборе команды Inputcolor открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – три поля ввода типа TEdit метками Red, Green, Blue; – группу из трех флажков (Left, Right) типа TCheckBox; – кнопку типа TButton. <p>Обеспечить возможность ввода RGB-составляющих цвета. При выборе команды Color, цвет фона главного окна изменяется на заданный (левая, правая или обе половины окна в зависимости от установки флажков).</p>
16.	<p>Написать приложение Windows, — графическую иллюстрацию вычисление корней уравнения методом деления отрезка пополам</p> <p>Создать меню с командами: Root, Show, About, Exit.</p> <p>Команда Exit завершает приложение. Команда About открывает окно с информацией о разработчике. Для выбора файла исходных данных (команда Open) использовать объект OpenFileDialog. Исходные данные для аппроксимации — массивы экспериментальных значений аргумента x и функции $y(x)$ — сформировать самостоятельно.</p> <p>При выборе команды Root выводится значение корня. При выборе команды Show в главном окне приложения отображается график функции и исходные данные в виде точек. Окно должно содержать заголовок. Изображение должно занимать все окно.</p>
17.	<p>Создать меню с командами: Input, Calc, Exit</p> <p>При выборе команды Input открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – три поля ввода типа TEdit;

	<ul style="list-style-type: none"> – группу из двух флажков («периметр», «площадь») типа <code>TCheckBox</code>; – кнопку типа <code>TButton</code> (свойство <code>Default</code> имеет значение <code>true</code>). <p>Обеспечить возможность:</p> <ul style="list-style-type: none"> – ввода длин сторон квадрата; <p>выбора режима с помощью флажков: подсчет периметра и (или) площади квадрата. При выборе команды Calc открывается диалоговое окно с результатами.</p>
18.	<p>Создать меню с командами: Input, Calc, Exit</p> <p>При выборе команды Input открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – три поля ввода типа <code>TEdit</code>; – группу из двух флажков («периметр», «площадь») типа <code>TCheckBox</code>; – кнопку типа <code>TButton</code> (свойство <code>Default</code> имеет значение <code>true</code>). <p>Обеспечить возможность:</p> <ul style="list-style-type: none"> – ввода длин сторон прямоугольника; <p>выбора режима с помощью флажков: подсчет периметра и (или) площади прямоугольника. При выборе команды Calc открывается диалоговое окно с результатами.</p>
19.	<p>Создать меню с командами: Input, Calc, Exit</p> <p>При выборе команды Input открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – три поля ввода типа <code>TEdit</code>; – группу из двух флажков («периметр», «площадь») типа <code>TCheckBox</code>; – кнопку типа <code>TButton</code> (свойство <code>Default</code> имеет значение <code>true</code>). <p>Обеспечить возможность:</p> <ul style="list-style-type: none"> – ввода длины радиуса окружности; <p>выбора режима с помощью флажков: подсчет периметра и (или) площади круга. При выборе команды Calc открывается диалоговое окно с результатами.</p>
20.	<p>Написать приложение Windows, которое по заданным в файле исходным данным строит круговую или столбиковую диаграмму. Создать меню с командами: Input data, Choose, Line, Bar, Quit.</p> <p>Команды Line, Bar запрещены. Команда Quit завершает приложение. При выборе команды Input data из файла читаются исходные данные (файл сформировать самостоятельно).</p> <p>По команде Choose открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – окно-список для выбора цвета графика типа <code>TListBox</code>; – группу из двух переключателей (Line, Bar) типа <code>TRadioGroup</code>; – кнопку типа <code>TButton</code>. <p>Обеспечить возможность ввода цвета и выбора режима: построение графика (Line) или столбиковой диаграммы (Bar). После указания параметров разрешается соответствующая команда меню.</p> <p>По команде Line или Bar в главном окне приложения выбранным цветом строится диаграмма одного или другого вида. Окно должно содержать заголовок графика или диаграммы, наименование и града-</p>

	<p>цию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна</p>
21.	<p>Написать приложение Windows, которое по заданным в файле исходным данным строит график или круговую диаграмму. Создать меню с командами: Input data, Choose, Line, Bar, Quit. Команды Line, Bar запрещены. Команда Quit завершает приложение. При выборе команды Input data из файла читаются исходные данные (файл сформировать самостоятельно).</p> <p>По команде Choose открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – окно-список для выбора цвета графика типа TListBox; – группу из двух переключателей (Line, Bar) типа TRadioGroup; – кнопку типа TButton. <p>Обеспечить возможность ввода цвета и выбора режима: построение графика (Line) или столбиковой диаграммы (Bar). После указания параметров разрешается соответствующая команда меню.</p> <p>По команде Line или Bar в главном окне приложения выбранным цветом строится график или диаграмма. Окно должно содержать заголовок графика или диаграммы, наименование и градацию осей. Изображение должно занимать все окно и масштабироваться при изменении размеров окна</p>
22.	<p>Создать меню с командами: Begin, Help, About. При выборе команды Begin открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – 2 поля ввода чисел X и Y типа TEdit сметкой Input; – метку типа TLabel для вывода результата; – группу из трех переключателей (+, /,-) типа TRadioGroup; – две кнопки типа TButton — Do и OK. <p>Обеспечить возможность:</p> <ul style="list-style-type: none"> – ввода числа в поле Input; – выбора режима преобразования с помощью переключателей: перевод в двоичную, шестнадцатеричную или восьмеричную систему счисления. <p>При щелчке на кнопке Do должен появляться результат расчета сложения, деления и умножения двух чисел</p>
23.	<p>Создать меню с командами: Begin, Help, About. При выборе команды Begin открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – поле ввода типа TEdit сметкой Input; – метку типа TLabel для вывода результата; – группу из трех переключателей (-, *, /) типа TRadioGroup; – две кнопки типа TButton — Do и OK. <p>Обеспечить возможность:</p> <ul style="list-style-type: none"> – ввода числа в поле Input;

	<ul style="list-style-type: none"> – выбора режима преобразования с помощью переключателей: перевод в двоичную, шестнадцатеричную или восьмеричную систему счисления. <p>При щелчке на кнопке До должен появляться результат расчета разности, умножения и деления двух чисел</p>
24.	<p>Создать меню с командами: Begin, Help, About.</p> <p>При выборе команды Begin открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – 2 поля ввода чисел X и Y типа TEdit сметкой Input; – метку типа TLabel для вывода результата; – группу из трех переключателей (+, -, *) типа TRadioGroup; – две кнопки типа TButton— Дои ОК. <p>Обеспечить возможность:</p> <ul style="list-style-type: none"> – ввода числа в поле Input; – выбора режима преобразования с помощью переключателей: перевод в двоичную, шестнадцатеричную или восьмеричную систему счисления. <p>При щелчке на кнопке До должен появляться результат расчета сложения, разности и умножения двух чисел</p>
25.	<p>Создать меню с командами: Begin, Help, About.</p> <p>При выборе команды Begin открывается диалоговое окно, содержащее:</p> <ul style="list-style-type: none"> – 2 поля ввода чисел X и Y типа TEdit сметкой Input; – метку типа TLabel для вывода результата; – группу из трех переключателей (+, *, /) типа TRadioGroup; – две кнопки типа TButton— Дои ОК. <p>Обеспечить возможность:</p> <ul style="list-style-type: none"> – ввода числа в поле Input; – выбора режима преобразования с помощью переключателей: перевод в двоичную, шестнадцатеричную или восьмеричную систему счисления. <p>При щелчке на кнопке До должен появляться результат вычисления суммы, произведения частного двух чисел</p>

6. Перечень основной и дополнительной учебной литературы

Основная литература:

1. Рик, Г. Объектно-Ориентированное Программирование / Г. Рик ; под редакцией Н. Комлева. — Москва : СОЛОН-Пресс, 2023. — 298 с

2. Григорьев А.А. Методы и алгоритмы обработки данных : учеб. Пособие — М.: ИНФРА-М, 2021. — 383 с. / ЭБС «Знаниум». Режим доступа: URL <https://znanium.com/read?id=361208>

Дополнительная литература:

1. Хорев П.Б. Объектно-ориентированное программирование с примерами на С#. М.: Форум. 2022. – 200 с. ЭБС «Знаниум». Режим доступа: URL <https://znanium.com/read?id=351782>
2. Ачкасов, В. Программирование на Lazarus / В. Ачкасов. - 2-е изд., исправ. - Москва: Национальный Открытый Университет «ИНТУИТ», 2022. - 521 с. Режим доступа: URL: <http://biblioclub.ru/index.php?page=book&id=429187>

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины
Интернет-ресурсы:

<http://www.biblioclub.ru>

<http://znanium.com>

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения:

Любая интегрированная среда программирования, поддерживающая язык Object Pascal, например, Lazarus (Delphi 7), MS Office 15.

Информационные справочные системы:

не предусмотрено курсом данной дисциплины.

Ресурсы информационно-образовательной среды ТУ.

Рабочая программа и методическое обеспечение по дисциплине «Объектно-ориентированное программирование».