

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Белгородский Валерий Савельевич  
Должность: Ректор  
Дата подписания: 24.06.2024 17:01:18  
Уникальный программный ключ:  
8df276ee93e17c18e7bee9e7cad2d0ed9abb2479

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Российский государственный университет им. А.Н. Косыгина  
(Технологии. Дизайн. Искусство)»

Институт информационных технологий и цифровой трансформации  
Кафедра информационных технологий

## РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

### Объектно-ориентированное программирование. Современные паттерны

Уровень образования	бакалавриат
Направление подготовки	09.03.02 Информационные системы и технологии
Профиль	Информационные технологии и дизайн
Срок освоения образовательной программы по очной форме обучения	4 года
Форма обучения	очная

Рабочая программа учебной дисциплины «Объектно-ориентированное программирование. Современные паттерны» основной профессиональной образовательной программы высшего образования рассмотрена и одобрена на заседании кафедры, протокол № 9 от 11.04.2024 г.

Разработчики рабочей программы «Объектно-ориентированное программирование. Современные паттерны»:

1. Канд. техн. наук, доцент А.А. Семёнов
2. Старший преподаватель А.М. Козлов

Заведующий кафедрой: канд. техн. наук, доцент И.Б. Разин

## **1. ОБЩИЕ СВЕДЕНИЯ**

Учебная дисциплина «Объектно-ориентированное программирование. Современные паттерны» изучается в шестом семестре.

Курсовая работа/Курсовой проект – не предусмотрены.

### **1.1. Форма промежуточной аттестации: экзамен.**

При проведении промежуточной аттестации применяется Методика использования балльно-рейтинговой системы при реализации основных профессиональных образовательных программ высшего образования Института информационных технологий и цифровой трансформации, подписанная 08.04.2024 директором ИИТиЦТ Чикуновым И.М.

### **1.2. Место учебной дисциплины в структуре ОПОП**

Учебная дисциплина «Объектно-ориентированное программирование. Современные паттерны» относится к части, формируемой участниками образовательных отношений.

Основой для освоения дисциплины являются результаты обучения по предшествующим дисциплинам и практикам:

- Программирование;
- Прикладное программирование;
- Алгоритмы и структуры данных.

Результаты обучения по учебной дисциплине используются при изучении следующих дисциплин:

- Программная инженерия и гибкие методологии разработки ПО;
- Сетевые технологии и программирование.

## **2. ЦЕЛИ И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ**

Целями изучения дисциплины «Объектно-ориентированное программирование. Современные паттерны» являются:

- изучение способов представления и структурирования информации о явлениях и процессах в окружающем мире применительно к своей профессиональной деятельности;
- освоение методов ориентирования и взаимодействия с ресурсами информационной среды, осуществления выбора различных моделей использования программных средств разработки для информационных и автоматизированных систем;
- изучение методов построения алгоритмов и основных этапов разработки и создания современных программных продуктов с учетом основных требований информационной безопасности;
- формирование навыков научно-практического подхода к построению эффективных диалоговых интерфейсов, ориентированных на пользователя;
- изучение базовых принципов, методов и средств решения стандартных задач современного объектно-ориентированного и визуального программирования;
- формирование у обучающихся компетенций, установленных образовательной программой в соответствии с ФГОС ВО по данной дисциплине.

Результатом обучения по учебной дисциплине является овладение обучающимися знаниями, умениями, навыками и опытом деятельности, характеризующими процесс формирования компетенций и обеспечивающими достижение планируемых результатов освоения учебной дисциплины.

2.1. Формируемые компетенции, индикаторы достижения компетенций, соотнесённые с планируемыми результатами обучения по дисциплине:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине
<p>ПК-2. Способен реализовывать проекты цифровой трансформации предприятий в самостоятельно выбранной предметной области, в том числе разрабатывать новые информационные и цифровые продукты путем применения существующих информационных и цифровых технологий, а также их адаптации под заданные условия, требования и ограничения</p>	<p>ИД-ПК-2.1 Определение принадлежности задачи профессиональной деятельности заданному классу и предметной области</p>	<ul style="list-style-type: none"> <li>– Формирует перечень задач в профессиональной деятельности для получения заданного результата (достижения заданной цели).</li> <li>– Самостоятельно использует методы и средства познания, обучения и самоконтроля для приобретения новых знаний и умений с использованием современных информационных технологий.</li> <li>– Оценивает предметную область с использованием сетевыми средствами для обмена данными в глобальной информационной сети Интернет.</li> <li>– Анализирует и обобщает информацию для правильной постановки цели и нахождения способов ее достижения.</li> </ul>
	<p>ИД-ПК-2.2 Выбор оптимального набора инструментальных средств и ИТ-методов решения профессиональной задачи в рамках предметной области</p>	<ul style="list-style-type: none"> <li>– Оценивает набор инструментальных средств решения профессиональных задач с учетом имеющихся ресурсов.</li> <li>– Оценивает качество ИТ-методов решения задач в соответствии с предметной областью.</li> <li>– Прогнозирует зависимость результата достижения цели от качества решения ИТ-задачи.</li> <li>– Самостоятельно использует типовые инструменты контроля решения ИТ-задач.</li> </ul>
	<p>ИД-ПК-2.3 Адаптация современных методов и алгоритмов под конкретные задачи выбранной предметной области</p>	<ul style="list-style-type: none"> <li>– Применяет методики использования программных средств для решения практических задач в информационных и автоматизированных системах.</li> <li>– Умеет разрабатывать современные эффективные интерфейсы «человек - электронно-вычислительная машина».</li> <li>– Анализирует навыки работы с программными средствами для управления информацией и коммуникации на основе базовых принципов современных информационных технологий.</li> <li>– Рационально оценивает и обосновывает принимаемые проектные решения для выбора и установки программных средств.</li> </ul>

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения по дисциплине
		<ul style="list-style-type: none"> <li>– Самостоятельно разрабатывает и составляет структурные схемы алгоритмов и реализует их на языках программирования высокого уровня.</li> </ul>
	ИД-ПК-2.4 Использование ИТ-инструментов для решения задачи в выбранной предметной области	<ul style="list-style-type: none"> <li>– Оценивает сущность и значение информационных технологий и инструментов для решения практических задач.</li> <li>– Проводит анализ основных методов, способов и средств получения, хранения, переработки, защиты и визуализации информации.</li> <li>– Самостоятельно осуществляет инсталляцию и практическую реализацию информационно-коммуникационных программ и компьютерных приложений.</li> <li>– Оценивает выполнение тестовых примеров для проверки их корректности и эффективности.</li> <li>– Анализирует методики использования программных средств для решения практических задач в информационных и автоматизированных системах.</li> </ul>

### 3. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

Общая трудоёмкость учебной дисциплины по учебному плану составляет:

по очной форме обучения –	6	з.е.	192	час.
---------------------------	---	------	-----	------

#### 3.1. Структура учебной дисциплины для обучающихся по видам занятий (очная форма обучения)

Структура и объем дисциплины				
Объем дисциплины по семестрам	ом еж уто чн ой	все го, час	Контактная аудиторная работа, час	Самостоятельная работа обучающегося, час

			лекции, час	практические занятия, час	лабораторные занятия, час	практическая подготовка, час	курсовая работа/ курсовой проект	самостоятельная работа обучающегося, час	промежуточная аттестация, час
6 семестр	экзамен	192	34		34	6		86	32
	Всего:	192	34		34	6		86	32

## 3.2. Структура учебной дисциплины для обучающихся по разделам и темам дисциплины: (очная форма обучения)

Планируемые (контролируемые) результаты освоения: код(ы) формируемой(ых) компетенции(й) и индикаторов достижения компетенций	Наименование разделов, тем; форма(ы) промежуточной аттестации	Виды учебной работы				Самостоятельная работа, час	Виды и формы контрольных мероприятий, обеспечивающие по совокупности текущий контроль успеваемости; формы промежуточного контроля успеваемости
		Контактная работа					
		Лекции, час	Практические занятия, час	Лабораторные работы/индивидуальные занятия, час	Практическая подготовка, час		
<b>Шестой семестр</b>							
ПК-2: ИД-ПК-2.1 ИД-ПК-2.2	<b>Раздел I. Языки программирования высокого уровня</b>	4	х	4	х	12	Формы текущего контроля по разделам: 1. Выполнение лабораторных работ. 2. Контроль посещений. 3. Посещение профориентационных мероприятий. 4. Участие (достижения) в профессиональных конкурсах. 5. Научная и/или практическая работа.
	Лекция 1.1. Структура программы на языках высокого уровня	2				х	
	Лекция 1.2. Основные операторы языков высокого уровня	2				х	
	Лабораторная работа № 1.1. Простые вычисления с числами целых типов			1		4	
	Лабораторная работа № 1.2. Разработка приложения с графическим интерфейсом			2		4	
	Лабораторная работа № 1.3. Использование свойств визуальных компонентов			1		4	
ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3	<b>Раздел II. Структурные типы данных</b>	4	х	4	1	12	
	Лекция 2.1. Символьные и строковые типы данных	1				х	
	Лекция 2.2. Тип-диапазон, перечисляемые типы, массивы	1				х	
	Лекция 2.3. Множества, записи	2				х	
	Лабораторная работа № 2.1. Работа с символьными данными			1		4	
	Лабораторная работа № 2.2. Работа с матрицами (двумерными массивами)			2		4	
	Лабораторная работа № 2.3. Работа с записями			1	1	4	
ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3	<b>Раздел III. Подпрограммы</b>	4	х	4	1	12	
	Лекция 3.1. Описание подпрограмм. Параметры подпрограмм	1				х	
	Лекция 3.2. Процедурный тип данных	1				х	

Планируемые (контролируемые) результаты освоения: код(ы) формируемой(ых) компетенции(й) и индикаторов достижения компетенций	Наименование разделов, тем; форма(ы) промежуточной аттестации	Виды учебной работы				Самостоятельная работа, час	Виды и формы контрольных мероприятий, обеспечивающие по совокупности текущий контроль успеваемости; формы промежуточного контроля успеваемости
		Контактная работа					
		Лекции, час	Практические занятия, час	Лабораторные работы/ индивидуальные занятия, час	Практическая подготовка, час		
ИД-ПК-2.4	Лекция 3.3. Структура модулей. Интерфейсная и исполняемая часть модулей	1				x	
	Лекция 3.4. Создание и использование библиотек DLL	1				x	
	Лабораторная работа № 3.1. Использование подпрограмм и модулей			2		6	
	Лабораторная работа № 3.2. Использование библиотек DLL			2	1	6	
ПК-2:	<b>Раздел IV. Файлы</b>	4	x	4	1	12	
ИД-ПК-2.1	Лекция 4.1. Способы доступа к различным типам файлов	2					
ИД-ПК-2.2	Лекция 4.2. Основные операторы для работы с различными типами файлов	2					
ИД-ПК-2.3	Лабораторная работа № 4.1. Поиск файлов. Работа с найденными текстовыми файлами			2		6	
ИД-ПК-2.4				2	1	6	
ПК-2:	<b>Раздел V. Основы объектно-ориентированного программирования</b>	4	x	4	1	12	
ИД-ПК-2.1	Лекция 5.1. Принципы объектно-ориентированного программирования	1				x	
ИД-ПК-2.2	Лекция 5.2. Объектные типы данных	1				x	
ИД-ПК-2.3		1				x	
ИД-ПК-2.4	Лекция 5.4. События и их обработка	1				x	
	Лабораторная работа № 5.1. Статическая графика в окнах Windows			2		6	

Планируемые (контролируемые) результаты освоения: код(ы) формируемой(ых) компетенции(й) и индикаторов достижения компетенций	Наименование разделов, тем; форма(ы) промежуточной аттестации	Виды учебной работы				Самостоятельная работа, час	Виды и формы контрольных мероприятий, обеспечивающие по совокупности текущий контроль успеваемости; формы промежуточного контроля успеваемости
		Контактная работа					
		Лекции, час	Практические занятия, час	Лабораторные работы/ индивидуальные занятия, час	Практическая подготовка, час		
	Лабораторная работа № 5.2. Динамическая графика в окнах Windows			2	1	6	
ПК-2: ИД-ПК-2.3 ИД-ПК-2.4	<b>Раздел VI. Использование динамической памяти</b>	6	x	6	1	12	
	Лекция 6.1. Использование указателей	2					
	Лекция 6.2. Выделение и освобождение динамической памяти	2					
	Лекция 6.3. Жизненный цикл программы	2					
	Лабораторная работа № 6.1. Использование указателей			6	1	12	
ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3 ИД-ПК-2.4	<b>Раздел VII. Технологии объектно-ориентированного программирования</b>	8	x	8	x	14	
	Лекция 7.1. Технологии ООП	2					
	Лекция 7.2. Структуры и динамические структуры данных	2					
	Лекция 7.3. Классы и современные паттерны ООП	4					
	Лабораторная работа № 7.1. Сортировки Яндекс. Оптимизация программного кода			2		4	
	Лабораторная работа № 7.2. Применение структур и ДСД			2		4	
	Лабораторная работа № 7.3. Классы. Инкапсуляция. Наследование и виртуальные методы			4	1	6	
	Экзамен	x	x	x	x	32	Компьютерный тест. Промежуточная аттестация производится в рамках балльно-рейтинговой системы. Оценка по дисциплине выставляется в соответствии с Системой оценивания результатов текущего контроля и



Планируемые (контролируемые) результаты освоения: код(ы) формируемой(ых) компетенции(й) и индикаторов достижения компетенций	Наименование разделов, тем; форма(ы) промежуточной аттестации	Виды учебной работы				Самостоятельная работа, час	Виды и формы контрольных мероприятий, обеспечивающие по совокупности текущий контроль успеваемости; формы промежуточного контроля успеваемости
		Контактная работа					
		Лекции, час	Практические занятия, час	Лабораторные работы/индивидуальные занятия, час	Практическая подготовка, час		
							промежуточной аттестации.
	<b>ИТОГО за шестой семестр</b>	<b>34</b>		<b>34</b>	<b>6</b>	<b>118</b>	<b>Экзамен</b>
	<b>ИТОГО за весь период</b>	<b>34</b>		<b>34</b>	<b>6</b>	<b>118</b>	

## 3.3. Краткое содержание учебной дисциплины

№ пп	Наименование раздела и темы дисциплины	Содержание раздела (темы)
<b>Раздел I</b>	<b>Языки программирования высокого уровня.</b>	
Лекция 1.1	Структура программы на языках высокого уровня	Введение. Основные понятия программирования. Структура программы на языках-компиляторах в общем виде с модулями, функциями и процедурами. Основные операторы. Отладка и компиляция программ. Интерпретация программ на языках высокого уровня. Примеры языков и сред разработки.
Лекция 1.2	Типы данных в языках высокого уровня. Скалярные числовые типы	Типы данных и их представление в двоичном коде. Различие операций с разными типами данных. Основные операторы и операции для работы с разными типами. Важность использования функций преобразования типов и форматирования в операциях ввода-вывода. Целесообразность и эффективность при выборе типов данных. Порядковые типы данных. Логические типы данных. Целые типы данных. Вещественные типы данных.
Лабораторная работа № 1.1	Простые вычисления с числами целых типов	Написание своего программного кода. Отладка и компиляция программ. Использование арифметических операций, процедур и функций для работы с целыми типами данных согласно вариантам заданий.
Лабораторная работа № 1.2	Разработка приложения с графическим интерфейсом	Создание современных прикладных программ методами визуального программирования. Конструктор формы и визуальные компоненты. Свойства и методы объектов. Создание приложения с графическим интерфейсом, ветвлением и циклом для решения задачи в соответствии с вариантом
Лабораторная работа № 1.3	Использование свойств визуальных компонентов	Создание приложения с графическим интерфейсом для перевода значений из одной системы счисления в другую в соответствии с вариантом задания. Перевод должен производиться по выбору пользователя. Расчёт по соответствующим формулам и вывод результатов.
<b>Раздел II</b>	<b>Структурные типы данных</b>	
Лекция 2.1	Символьные и строковые типы данных	Однобайтные и двухбайтные символьные данные. Операции с символьными данными. Связь символов и их кодов в различных кодовых таблицах. Строковые типы данных как массивы символов. Операторы для работы со строками. Преобразования строк разных типов.
Лекция 2.2	Тип-диапазон, перечисляемые типы, массивы	Использование типа-диапазона в операторах множественного выбора и при объявлении массивов. Применение перечисляемых типов данных для работы с ассоциативными массивами. Массивы как способ создания последовательных списков однородных данных. Диапазоны индексов массивов. Обращение к элементам массивов. Особенности использования одномерных и многомерных массивов.
Лекция 2.3	Множества, записи	Особенности операций с типом-множество. Количество возможных присваиваемых значений переменным типа множество. Примеры программ, использующих тип-множество. Записи как основной способ объединения и структурирования данных разных типов. Объявление записей и их использование в программах. Доступ к полям

		записей. Вариантные поля записей.
Лабораторная работа № 2.1	Работа с символьными данными.	Обработка символьных строк в соответствии с вариантами заданий. Анализ и преобразования символьных данных.
Лабораторная работа № 2.2	Работа с матрицами (двумерными массивами).	Изучение методик алгоритмизации при работе с многомерными массивами: индексация элементов, поиск элементов в строках и столбцах. Ввод данных в двумерный массив (матрицу). Обработка данных (математические вычисления в строках и столбцах матрицы) в соответствии с вариантами заданий.
Лабораторная работа № 2.3	Работа с записями.	Изучение методик алгоритмизации при работе с записями: разработка рациональных структурированных данных с разнотипными полями, создание списков записей, обработка записей для составления выборок информации. Ввод структурированной информации в поля записей. Создание выборок информации по заданным полям в соответствии с вариантами заданий и вывод результатов вычислений.
<b>Раздел III</b>	<b>Подпрограммы</b>	
Лекция 3.1	Описание подпрограмм. Параметры подпрограмм	Описание подпрограмм в соответствующих разделах языка высокого уровня. Параметры подпрограмм: параметры-значения, параметры-переменные, параметры-константы. Выбор способа передачи параметров в подпрограмму и возвращения результатов. Различные способы вызова подпрограмм. Рекурсия и опережающее описание.
Лекция 3.2	Процедурный тип данных	Создание и использование процедурных типов данных. Тип-процедура и тип-функция. Передача подпрограммы в качестве параметра в другую подпрограмму. Вызов внешних подпрограмм из модулей и библиотек. Адрес подпрограммы и его получение для доступа и вызова подпрограммы.
Лекция 3.3	Структура модулей. Интерфейсная и исполняемая часть модулей	Модульная структура приложений. Понятие «проект» в современных средах разработки программ на языках высокого уровня. Создание файла-модуля. Структура модуля. Интерфейсная и исполняемая часть модулей. Доступ к объявленным в модуле подпрограммам. Подключение модулей к головной программе и вызов подпрограмм.
Лекция 3.4	Создание и использование библиотек DLL	Способы создания файлов на языках высокого уровня для их компиляции в динамически подключаемые библиотеки. Особенности использования библиотек в различных операционных системах. Статический и динамический способы вызова подпрограмм из библиотек. Основные операторы для работы с DLL в операционных системах семейства Windows.
Лабораторная работа № 3.1	Использование подпрограмм и модулей	Разработка структуры программы с процедурами и функциями. Оформление подпрограммы в отдельном модуле. Подключение модуля к головной программе и использование подпрограммы. Тип подпрограммы и способ обработки данных указаны в вариантах заданий.

Лабораторная работа № 3.2	Использование библиотек DLL.	Оформление подпрограммы с заданиями по вариантам в библиотеках DLL. Вызов подпрограмм из головной программы статическим и динамическим способами.
<b>Раздел IV</b>	<b>Файлы</b>	
Лекция 4.1	Способы доступа к различным типам файлов	Понятие о файловых системах как способах структурирования данных. Объявление и доступ к различным типам файлов. Связь понятий «файлы» и «поток». Имена файлов и логические устройства. Поиск файлов на жестком диске по имени и расширению. Системные функции для работы с файлами. Иерархическая древовидная структура файловой системы в современных операционных системах. Свойства файлов: имя, тип, атрибуты, размер, дата создания/изменения.
Лекция 4.2	Основные операторы для работы с различными типами файлов	Текстовые, типизированные, нетипизированные файлы – различные способы инициализации для чтения и записи. Последовательный и прямой доступ к данным. Особенности работы с файлами разных типов. Различные операторы для чтения и записи данных. Определение признака конца строки в текстовых файлах последовательного доступа. Перемещение файлового указателя в файлах прямого доступа. Определение признака конца файла.
Лабораторная работа № 4.1	Поиск файлов. Работа с найденными текстовыми файлами.	Ввод произвольного имени текстового файла. Поиск на жестком диске компьютера всех файлов с заданным именем (одноименные файлы могут находиться в разных папках), поочередный вывод на экран их содержимого. Выполнение обработки каждого найденного файла в соответствии с заданием в варианте.
Лабораторная работа № 4.2	Обработка данных в типизированных файлах.	Разработка алгоритма и прикладной программы обработки типизированных файлов и анализа данных в их содержимом.
<b>Раздел V</b>	<b>Основы объектно-ориентированного программирования</b>	
Лекция 5.1	Принципы объектно-ориентированного программирования	Основные понятия и принципы объектно-ориентированного программирования. Инкапсуляция: информация об объекте (его свойства) и функции объекта (методы) содержатся в описании этого объекта. Наследование: один объект может быть построен на базе описания другого объекта. Полиморфизм: способность многих объектов использовать один и тот же метод, при этом выполняемые действия зависят от объекта, который его вызвал. Применение этих принципов при разработке современных программных продуктов.
Лекция 5.2	Общая характеристика объектов и классов	Структуры объектов и классов в объектно-ориентированных языках высокого уровня. Свойства и методы объектов и классов. Конструктор и деструктор объекта. Объявление объектов в разделе описаний и создание методов в виде подпрограмм. Инициализация полей объектов. Статическое и динамическое создание экземпляров объектов и классов. Использование объектов и классов в современных средах разработки приложений. Библиотеки объектов, классов.

Лекция 5.3	Использование визуальных компонентов современных сред программирования	Понятия форм и окон в графических операционных системах. Визуальные компоненты в современных средах разработки приложений с графическими интерфейсами. Интерфейс с графическим устройством – GDI. Основные изобразительные средства операционных систем семейства Windows: перо, кисть, шрифт. Свойства основных изобразительных средств и их использование в прикладных программах.
Лекция 5.4	События и их обработка	Возникновение и классификация событий. События, генерируемые операционной системой. События, генерируемые программами и процессами. Воздействия пользователя на программу посредством генерирования событий текстовым устройством ввода (клавиатурой) и графическими указателями (мышь, джойстик, палец на интерактивном экране мобильного смартфона и т.п.). Особенности обработки различных типов событий в прикладных программах.
Лабораторная работа № 5.1	Статическая графика в окнах Windows.	Создать окна в операционной системе Windows. Используя функции из библиотеки Lazarus и GDI API Windows, создать в окне Windows изображение по заданию в варианте. В изображении обязательно хотя бы один раз использовать линии, прямоугольники, окружности, многоточечные фигуры-полигоны. Обязательно использовать разные типы и цвета линий, разные типы заливки фигур. Обязательно использовать текстовые надписи.
Лабораторная работа № 5.2	Динамическая графика в окнах Windows.	Создание окна Windows. Разделение окна линиями на заданное в варианте количество частей. Линии должны изменяться в соответствии с изменениями размеров окна воздействиями пользователя. Написать процедуры для обработки нажатия на клавиши мыши по заданию в варианте. Геометрические фигуры должны иметь центр в точке нажатия «мыши», текстовые сообщения должны выводиться, начиная от точки нажатия «мыши». При перерисовке окна (например, во время изменения его размеров), созданные с помощью «мыши» изображения фигур и текстовые надписи должны сохраняться и обновляться.
<b>Раздел VI</b>	<b>Использование динамической памяти</b>	
Лекция 6.1	Использование указателей	Понятия об указателях и ссылочной системе адресации данных. Различные типы данных имеют различные размеры областей хранения. Типизированные и нетипизированные указатели. Определение адреса хранящейся в памяти области данных. Специфика обращения по адресам указателей и ссылкам к разным типам данных. Возможные операции с указателями. Операции с данными, расположенными по адресам указателей.
Лекция 6.2	Выделение и освобождение динамической памяти	Разные способы выделения и освобождения динамической памяти для типизированных и нетипизированных указателей. Функции языков высокого уровня выделения и

		освобождения памяти для разных типов указателей. Определение количества резервируемой памяти для нетипизированных указателей. Создание и использование динамических структур данных в памяти: списков, стеков, очередей, деков.
Лекция 6.3	Жизненный цикл программы	Применение основ объектно-ориентированного программирования для создания многозадачной системы с интерактивным интерфейсом. Основные принципы создания современных графических интерфейсов для удобной и эффективной интерактивной работы пользователей. Usability-технологии при разработке интерфейсов. Методы и способы создания современных конкурентоспособных продуктов для развития отечественного рынка прикладных программ.
Лабораторная работа № 6.1	Использование указателей	Выполнение лабораторной работы. Применение объектно-ориентированного программирования и визуальных средств разработки программ с использованием динамической памяти и указателей. Методы использования указателей и динамических структур данных.
<b>Раздел VII</b>	<b>Технологии объектно-ориентированного программирования</b>	
Лекция 7.1	Технологии ООП	Интегрированная среда разработки как инструмент для создания приложений. Создание приложений без интегрированных сред приложений. Характеристика основных технологий программирования. Структуры данных. Правила кодирования, документирования и основные этапы создания программного обеспечения.
Лекция 7.2	Структуры и динамические структуры данных	Структуры, структуры и функции, массивы структур, поиск в массиве структур, вложенность структур, рекурсия, алгоритмы быстрых сортировок, массивы структур и бинарные файлы. Оптимизация кода и абстракции. Понятие и предназначение динамических структур данных (ДСД). Характеристика и синтаксис таких ДСД, как линейные списки, стеки, очереди и бинарные деревья. Возможные области применения и операции над ДСД. Адаптеры и представления.
Лекция 7.3	Классы и современные паттерны ООП	Основные свойства ООП. Понятие и элементы класса. Отличие структур от классов. Спецификаторы доступа. Методы класса, отличие методов от функций. Доступ к методам класса. Конструкторы. Определение методов класса вне класса. Классы и память. Статические данные класса. Константные методы. Деструкторы. Синтаксис и принцип работы операторных функций. Правила перегрузки. Применение наследования при первоначальной разработке объектно-ориентированной программы. Иерархия классов. Общее и частное наследование, комбинации доступа. Множественное наследование. Виртуальные и дружественные функции.
Лабораторная работа № 7.1	Сортировки Яндекса. Оптимизация программного кода	Лабораторная работа посвящена изучению основных алгоритмов сортировок и способов их реализации на примере Яндекса (см. <a href="https://academy.yandex.ru/posts/osnovnyye-vidy-sortirovok-i-primery-ikh-realizatsii">https://academy.yandex.ru/posts/osnovnyye-vidy-sortirovok-i-primery-ikh-realizatsii</a> ). Работу необходимо реализовать в

		<p>виде визуального приложения. Нужно изучить представленный иллюстрированный пример реализации на C++ основных видов сортировок. Необходимо воспроизвести реализацию проекта и по аналогии доделать пирамидальную сортировку и сортировку слиянием. Выполнить оптимизацию кода и абстракцию. Разработать визуальное приложение, позволяющее визуализировать процесс сортировки целочисленного массива (вектора) из 12 случайных элементов с анимацией. Допускается использовать подходящий стандартный компонент, либо конструкцию из прямоугольников. Визуализируемую сортировку пользователь выбирает в списке (ComboBox). Замедление просто реализовать с помощью связки sleep() + ProcessMessages(). Выполнить оптимизацию кода и абстракцию.</p>
Лабораторная работа № 7.2	Применение структур и ДСД	<p>Лабораторная работа посвящена изучению структур и ДСД. Реализовать взаимодействия с геометрическими примитивами и отработать структуры. Создать визуальное приложение согласно представленному интерфейсу. Создать приложение, демонстрирующее работу списков, очередей и стеков. 1) Организовать считывание текстового файла с помощью структуры стек. 2) В л/р про авиарейсы нужно добавить поле дата вылета. Соответственно, при добавлении рейса необходимо формировать структуру очередь, согласно датам вылета самолетов. Нужно создать кнопку "Очередность вылетов", которая выведет созданную структуру. Создать визуальное приложение согласно представленному интерфейсу. Написать программу учёта для автосервиса, выполняющего кузовные работы и ремонт двигателей. При записи на обслуживание заполняется заявка, в которой указываются фамилия владельца, марка автомобиля, вид работы, дата приёма заказа и стоимость ремонта. После выполнения работы распечатывается квитанция (сохранение заявки в файл). Для хранения информации использовать динамическую структуру данных – очередь. Интерфейс программы организовать следующим образом: 1) добавление заявки; 2) распечатка квитанции (сохранение в файл); 3) вывод списка заявок на экран; 4) запись списка заявок в файлы (1 – кузовные работы; 2 – ремонт двигателя); 5) выход. Программу написать с использованием динамической структуры данных "Очередь" и компонента StringGrid.</p>
Лабораторная работа № 7.3	Классы. Инкапсуляция. Наследование и виртуальные методы	<p>Лабораторная работа посвящена изучению классов в ООП. Необходимо изучить и повторить пример, демонстрирующий применение объектно-ориентированной технологии наследования с виртуальными методами. Используя известный по лекциям класс <code>Tovar</code>, в примере создаётся два производных от него класса: 1) <code>TovarProd</code>, добавляющий возможность хранить информацию о сроке хранения и температуре хранения продуктовых товаров; 2) <code>TovarProm</code>, позволяющий хранить информацию в соответствии с полями базового класса. Необходимо создать улучшенную версию проекта, которая позволит работать</p>

		не с двумя массивами, а с одним универсальным, который может хранить данные как продуктовых товаров, так и промышленных. Также реализуем сохранение данных в текстовый файл и считывание их из него в компонент StringGrid. Также реализуем нормальное функционирование кнопки "Общая стоимость", которая сможет рассчитывать общую стоимость как добавленных вручную товаров, так и загруженных из файла. Реализовав рассмотренный пример, необходимо добавить возможности: а) сортировки товаров по наименованию и цене; б) поиска (по всем свойствам товаров).
--	--	---

### 3.4. Организация самостоятельной работы обучающихся

Самостоятельная работа студента – обязательная часть образовательного процесса, направленная на развитие готовности к профессиональному и личностному самообразованию, на проектирование дальнейшего образовательного маршрута и профессиональной карьеры.

Самостоятельная работа обучающихся по дисциплине организована как совокупность аудиторных и внеаудиторных занятий и работ, обеспечивающих успешное освоение дисциплины.

Аудиторная самостоятельная работа обучающихся по дисциплине выполняется на учебных занятиях под руководством преподавателя и по его заданию. Аудиторная самостоятельная работа обучающихся входит в общий объем времени, отведенного учебным планом на аудиторную работу, и регламентируется расписанием учебных занятий.

Внеаудиторная самостоятельная работа обучающихся – планируемая учебная, научно-исследовательская, практическая работа обучающихся, выполняемая во внеаудиторное время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия, расписанием учебных занятий не регламентируется.

Внеаудиторная самостоятельная работа обучающихся включает в себя:

- подготовку к лекциям, лабораторным работам и экзамену;
- изучение специальной рекомендованной литературы;
- изучение разделов/тем, не выносимых на лекции и практические занятия самостоятельно;
- подготовка к выполнению лабораторных работ;
- участие в рекомендованных контрольно-рейтинговых мероприятиях, в том числе профориентационных;
- подготовка к промежуточной аттестации в течение семестра.

Самостоятельная работа обучающихся с участием преподавателя в форме иной контактной работы предусматривает групповую и (или) индивидуальную работу с обучающимися и включает в себя:

- проведение индивидуальных и групповых консультаций по отдельным темам/разделам дисциплины;
- проведение консультаций перед экзаменом;
- консультации по организации самостоятельного изучения отдельных разделов/тем, базовых понятий учебных дисциплин профильного/родственного бакалавриата, которые формировали ОПК и ПК, в целях обеспечения преемственности образования.

Перечень разделов/тем, полностью или частично отнесенных на самостоятельное изучение с последующим контролем:



№ пп	Наименование раздела /темы дисциплины, выносимые на самостоятельное изучение	Задания для самостоятельной работы	Виды и формы контрольных мероприятий (учитываются при проведении текущего контроля)	Трудоемкость, час
<b>Раздел I Языки программирования высокого уровня</b>				
Лабораторная работа № 1.1	Простые вычисления с числами целых типов	Изучение учебной, научной и технической литературы по теме лабораторной работы. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения.	Выполнение лабораторной работы.	4
Лабораторная работа № 1.2	Разработка приложения с графическим интерфейсом	Изучение учебной, научной и технической литературы по теме лабораторной работы. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения.	Выполнение лабораторной работы.	4
Лабораторная работа № 1.3	Использование свойств визуальных компонентов	Изучение учебной, научной и технической литературы по теме лабораторной работы. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения.	Выполнение лабораторной работы.	4
<b>Раздел II Структурные типы данных</b>				
Лабораторная работа № 2.1	Работа с символьными данными.	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.	Выполнение лабораторной работы.	4
Лабораторная работа № 2.2	Работа с матрицами (двумерными массивами).	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов	Выполнение лабораторной работы.	4

		языка для выполнения задания лабораторной работы.		
Лабораторная работа № 2.3	Работа с записями.	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.	Выполнение лабораторной работы.	4
<b>Раздел III</b>	<b>Подпрограммы</b>			
Лабораторная работа № 3.1	Использование подпрограмм и модулей	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.	Выполнение лабораторной работы.	6
Лабораторная работа № 3.2	Использование библиотек DLL.	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.	Выполнение лабораторной работы.	6
<b>Раздел IV</b>	<b>Файлы</b>			
Лабораторная работа № 4.1	Поиск файлов. Работа с найденными текстовыми файлами.	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования.	Выполнение лабораторной работы.	6

		Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.		
Лабораторная работа № 4.2	Обработка данных в типизированных файлах.	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.	Выполнение лабораторной работы.	6
<b>Раздел V</b>	<b>Основы объектно-ориентированного программирования</b>			
Лабораторная работа № 5.1	Статическая графика в окнах Windows.	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.	Выполнение лабораторной работы.	6
Лабораторная работа № 5.2	Динамическая графика в окнах Windows.	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.	Выполнение лабораторной работы.	6
<b>Раздел VI</b>	<b>Использование динамической памяти.</b>			
Лабораторная работа № 6.1.	Использование указателей	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов	Выполнение лабораторной работы.	12

		объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.		
<b>Раздел VII</b>	<b>Технологии объектно-ориентированного программирования</b>			
Лабораторная работа № 7.1	Сортировки Яндекса. Оптимизация программного кода	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.	Выполнение лабораторной работы.	4
Лабораторная работа № 7.2	Применение структур и ДСД	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.	Выполнение лабораторной работы.	4
Лабораторная работа № 7.3	Классы. Инкапсуляция. Наследование и виртуальные методы	Изучение научной и технической литературы, нормативных документов, стандартов языков программирования. Работа с материалами конспекта лекций. Анализ задания к лабораторной работе, выбор способов её выполнения. Осваивание методов объектно-ориентированного и визуального программирования. Изучение элементов системы разработки программ и операторов языка для выполнения задания лабораторной работы.	Выполнение лабораторной работы.	6

### 3.5. Применение электронного обучения, дистанционных образовательных технологий

При реализации программы учебной дисциплины возможно применение электронного обучения и дистанционных образовательных технологий.

Реализация программы учебной дисциплины с применением электронного обучения и дистанционных образовательных технологий регламентируется действующими локальными актами университета.

Применяются следующие разновидности реализации программы с использованием ЭО и ДОТ.

В электронную образовательную среду, по необходимости, могут быть перенесены отдельные виды учебной деятельности:

<b>использование ЭО и ДОТ</b>	<b>использование ЭО и ДОТ</b>	<b>объем, час</b>	<b>включение в учебный процесс</b>
смешанное обучение	лекции	34	в соответствии с расписанием учебных занятий
	лабораторные занятия	34	

#### 4. РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ, КРИТЕРИИ ОЦЕНКИ УРОВНЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ, СИСТЕМА И ШКАЛА ОЦЕНИВАНИЯ

##### 4.1. Соотнесение планируемых результатов обучения с уровнями сформированности компетенций.

Итоговое количество баллов в 100-балльной системе по результатам текущей и промежуточной аттестации определяется в соответствии с Методикой использования балльно-рейтинговой системы при реализации основных профессиональных образовательных программ высшего образования Института информационных технологий и цифровой трансформации.

Уровни сформированности компетенции(-й)	Итоговое количество баллов в 100-балльной системе по результатам текущей и промежуточной аттестации	Оценка в пятибалльной системе по результатам текущей и промежуточной аттестации	Показатели уровня сформированности		
			универсальной(-ых) компетенции(-й)	общепрофессиональной(-ых) компетенций	профессиональной(-ых) компетенции(-й)
					ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3 ИД-ПК-2.4
высокий	85 – 100	отлично/ зачтено (отлично)/ зачтено			Обучающийся: – исчерпывающе и логически стройно излагает учебный материал, умеет связывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения по использованию информационных технологий; – способен уверенно использовать современные системы разработки прикладных программ с эффективными графическими интерфейсами и системы коммуникации в сети

					<p>Internet;</p> <ul style="list-style-type: none"> <li>– показывает творческие способности в понимании и практическом использовании языков высокого уровня, использовании визуальных компонентов разработки приложений графических интерфейсов;</li> <li>– дополняет теоретическую информацию сведениями, самостоятельно полученными из источников научно-технической информации;</li> <li>– способен провести целостный анализ среды разработки современных программ на основе объектно-ориентированного и визуального программирования;</li> <li>– свободно ориентируется в учебной и профессиональной литературе;</li> <li>– дает развернутые, исчерпывающие, профессионально грамотные ответы на вопросы, в том числе, дополнительные.</li> </ul>
повышенный	70 – 84	хорошо/ зачтено (хорошо)/ зачтено			<p>Обучающийся:</p> <ul style="list-style-type: none"> <li>– достаточно подробно, грамотно и по существу излагает изученный материал, приводит и раскрывает в тезисной форме основные понятия информационных технологий;</li> <li>– анализирует современные</li> </ul>

					<p>технология программирования с незначительными пробелами;</p> <ul style="list-style-type: none"> <li>– способен использовать только основные функциональные возможности систем разработки программ и систем коммуникации в сети Internet;</li> <li>– способен провести анализ основных элементов разработки современных программ на основе объектно-ориентированного и визуального программирования;</li> <li>– допускает единичные негрубые ошибки;</li> <li>– достаточно хорошо ориентируется в учебной и профессиональной литературе;</li> <li>– ответ отражает знание теоретического и практического материала, не допуская существенных неточностей.</li> </ul>
базовый	55 – 69	удовлетворительно/ зачтено (удовлетворительно)/ зачтено			<p>Обучающийся:</p> <ul style="list-style-type: none"> <li>– демонстрирует теоретические знания основного учебного материала дисциплины в объеме, необходимом для дальнейшего освоения ОПОП;</li> <li>– с неточностями излагает принципы и методы разработки современных программ на основе объектно-ориентированного и визуального программирования;</li> <li>– способен использовать отдельные элементы визуальной</li> </ul>



					<p>разработки прикладных программ;</p> <ul style="list-style-type: none"> <li>– анализирует современные технологии программирования с неточностями и ошибками;</li> <li>– демонстрирует фрагментарные знания основной учебной литературы по дисциплине;</li> <li>– ответ отражает знания на базовом уровне теоретического и практического материала в объеме, необходимом для дальнейшей учебы и предстоящей работы по профилю обучения.</li> </ul>
низкий	0 – 54	неудовлетворительно/ не зачтено	<p>Обучающийся:</p> <ul style="list-style-type: none"> <li>– демонстрирует фрагментарные знания теоретического и практического материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации;</li> <li>– испытывает серьезные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами;</li> <li>– не способен проанализировать учебно-методическую, техническую и научную литературу;</li> <li>– не владеет основными принципами и навыками работы в современных средах разработки прикладных программ, не умеет пользоваться системами коммуникации (Internet);</li> <li>– выполняет задания только по образцу и под руководством преподавателя;</li> <li>– ответ отражает отсутствие знаний на базовом уровне теоретического и практического материала в объеме, необходимом для дальнейшей учебы.</li> </ul>		

## **5. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ, ВКЛЮЧАЯ САМОСТОЯТЕЛЬНУЮ РАБОТУ ОБУЧАЮЩИХСЯ**

При проведении контроля самостоятельной работы обучающихся, текущего контроля и промежуточной аттестации по учебной дисциплине «Объектно-ориентированное программирование. Современные паттерны» проверяется уровень сформированности у обучающихся компетенций и запланированных результатов обучения по дисциплине, указанных в разделе 2 настоящей программы.

## 5.1. Формы текущего контроля успеваемости, примеры типовых заданий:

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
1	Выполнение лабораторной работы. Лабораторная работа № 1.1	<p>Простые вычисления с числами целых типов. Создать консольную программу для решения задачи в соответствии с вариантом. Значения переменных для вычислений должны вводиться с клавиатуры (например, посредством операторов Write('n='); Readln(n);), а результат выводиться в консольную строку оператором Writeln(&lt;результат&gt;). Варианты:</p> <ol style="list-style-type: none"> <li>1. Дано целое число N (<math>0 &lt; N &lt; 1000</math>), определяющее возраст черепахи в годах. В зависимости от введённого значения грамотно сформировать фразу: «Черепаше &lt;N&gt; (год/года/лет)».</li> <li>2. Дано целое число N (<math>0 &lt; N &lt; 1000</math>), определяющее количество пойманных рыб. В зависимости от введённого значения грамотно сформировать фразу: «Поймано &lt;N&gt; (рыб/рыбы/рыба)».</li> <li>3. Дано целое число N (<math>0 &lt; N &lt; 1000</math>), определяющее количество найденных грибов. В зависимости от введённого значения грамотно сформировать фразу: «Найдено &lt;N&gt; (грибов/гриба/гриб)».</li> <li>4. Дано целое число N (<math>0 &lt; N &lt; 1000</math>), определяющее количество учеников в классе. В зависимости от введённого значения грамотно сформировать фразу: «В классе &lt;N&gt; (ученик/ученика/учеников)».</li> <li>5. Дано целое число N (<math>0 &lt; N &lt; 1000</math>), определяющее количество участников соревнований. В зависимости от введённого значения грамотно сформировать фразу: «Соревновалось &lt;N&gt; (участник/участника/участников)».</li> </ol> <p>...</p>	ПК-2: ИД-ПК-2.1
	Выполнение лабораторной работы. Лабораторная работа № 1.2	<p>Разработка приложения с графическим интерфейсом. Создать приложение с графическим интерфейсом для решения задачи в соответствии с вариантом. Значения переменных для вычислений должны вводиться в редактируемых элементах типа TEdit, а результат выводиться в текстовое поле типа TМето. В заголовок окна приложения записать свои данные: фамилию, имя и номер варианта. Варианты:</p> <ol style="list-style-type: none"> <li>1. Дано целое число N. Посчитать и вывести количество и сумму его цифр.</li> <li>2. Возвести целое число N в квадрат без операции умножения, используя</li> </ol>	ПК-2: ИД-ПК-2.1 ИД-ПК-2.2

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		<p>свойство: квадрат числа N равен сумме N первых нечетных чисел. Вывести все нечетные числа, используемые для вычислений.</p> <p>3. Разложить введенное с клавиатуры произвольное целое число N на все возможные целые простые множители. (Например: <math>20 = 2*2*5</math>; <math>21 = 3*7</math>; <math>22 = 2*11</math>; <math>23 = 1*23</math>).</p> <p>4. Для введенного с клавиатуры произвольного целого числа N определить и вывести все возможные целые делители, кроме 1 и самого числа. Если число не имеет таких делителей, сообщить об этом. (Например: для 20 – 2,4,5,10; для 21 – 3,7; для 23 – «нет делителей»).</p> <p>5. Даны целые числа N и M. Найти на интервале &lt;от меньшего из них до большего из них&gt; все целые числа, у которых результат деления на 2 является целым четным числом. Вывести все эти числа.</p> <p>...</p>	
	<p>Выполнение лабораторной работы. Лабораторная работа № 1.3</p>	<p>Использование свойств визуальных компонентов. Создать приложение с графическим интерфейсом для перевода значений из одной системы счисления в другую в соответствии с вариантом задания. Варианты:</p> <ol style="list-style-type: none"> <li>1. Разработать приложение, которое по выбору пользователя переводит значение температуры из шкалы Цельсия в шкалу Фаренгейта или наоборот.</li> <li>2. Разработать приложение, которое по выбору пользователя переводит значение температуры из шкалы Цельсия в шкалу Реомюра или наоборот.</li> <li>3. Разработать приложение, которое по выбору пользователя переводит значение расстояния из километров в сухопутные мили или наоборот.</li> <li>4. Разработать приложение, которое по выбору пользователя переводит значение расстояния из километров в морские мили или наоборот.</li> <li>5. Разработать приложение, которое по выбору пользователя переводит значение длины из метров в футы или наоборот.</li> </ol> <p>...</p>	<p>ПК-2: ИД-ПК-2.1 ИД-ПК-2.2</p>
	<p>Выполнение лабораторной работы. Лабораторная работа № 2.1</p>	<p>Работа с символьными данными. Создать приложение с графическим интерфейсом для решения задачи в соответствии с вариантом задания. Исходные строки должны вводиться в редактируемом строковом поле типа TМетод. В каждой строке имеются группы символов («подстроки»), которые разделяются одним или несколькими</p>	<p>ПК-2: ИД-ПК-2.3 ИД-ПК-2.4</p>

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		<p>пробелами. Пробелы могут находиться в начале и в конце строки. Подстрока или группа символов – это последовательность букв, цифр или знаков, отделенная пробелами от других групп, или находящаяся в начале/конце строки.</p> <p>В результирующее другое поле типа TМето должны быть перенесены (и/или преобразованы) только те строки, которые соответствуют условиям в вариантах заданий.</p> <p>Варианты:</p> <ol style="list-style-type: none"> <li>1. Симметричные группы символов из строки, если они состоят только из цифр. Например, строка: ‘ !2! 12321 reter 123 508805 1w1’; результат: ‘12321 508805’</li> <li>2. Симметричные группы символов из строки, если они состоят только из латинских букв. Например, строка: ‘ a11a abba 787 typeryt 1qq1’; результат: ‘abba typeryt’;</li> <li>3. Группы символов, состоящие только из латинских букв, если символы расположены в алфавитном порядке по возрастанию. Например, строка: ‘ 12*er faw acf ab2c fghjk’; результат: ‘acf fghjk’;</li> <li>4. Группы символов, состоящие только из латинских букв, если символы расположены в алфавитном порядке по убыванию. Например, строка: ‘ 12*er yfa y1fa acf ab2c ca’; результат: ‘yfa ca’;</li> <li>5. Группы символов, состоящие только из цифр, если цифры расположены по возрастанию. Например, строка: ‘ 1276 357 12e6 0279 ’; результат: ‘357 0279’;</li> </ol> <p>...</p>	
	<p>Выполнение лабораторной работы. Лабораторная работа № 2.2</p>	<p>Работа с матрицами (двумерными массивами). Создать приложение с графическим интерфейсом, производящее вычисления и обработку данных в соответствии с вариантом задания. Размерность обрабатываемой квадратной матрицы должна быть задана в редактируемом поле TEdit и не превышать количество зарезервированной размерности массива. Исходные данные должны вводиться в двумерный массив и отображаться в редактируемой таблице типа TStringGrid. Они могут вводиться ручным способом или через датчик случайных чисел на интервале &lt;-100..100&gt;. Введённые данные должны быть обработаны согласно варианту задания.</p>	<p>ПК-2: ИД-ПК-2.3 ИД-ПК-2.4</p>

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		<p>Преобразованный двумерный массив должен быть выведен в итоговую таблицу типа TStringGrid. Если результат является одномерным массивом (вектором), то он может быть выведен в текстовое поле типа TMemo. Скалярные (имеющие одно значение) результаты расчётов и вычислений могут быть выведены в виде текстовых ответов TLabel.</p> <p>Если результатами вычислений являются вещественные числа, то выводить их с точностью до 1-го знака после точки.</p> <p>Варианты:</p> <ol style="list-style-type: none"> <li>1. Вычислить среднее арифметическое элементов каждого из четных столбцов исходной матрицы.</li> <li>2. Поменять местами строку, содержащую максимальный элемент, со строкой, содержащей минимальный элемент. Если оба таких элемента находятся в одной строке, то вывести сообщение и номер строки.</li> <li>3. Получить новую матрицу вещественного типа путем деления всех элементов исходной матрицы на ее наименьший по модулю элемент (если он не равен 0).</li> <li>4. Получить новую матрицу вещественного типа путем умножения всех элементов исходной матрицы на ее наибольший по модулю элемент (если он не равен 0).</li> <li>5. Вычислить среднее арифметическое модулей элементов каждого из нечетных столбцов матрицы.</li> </ol> <p>...</p>	
	<p>Выполнение лабораторной работы. Лабораторная работа № 2.3</p>	<p>Работа с записями. Создать приложение с графическим интерфейсом, производящее вычисления и обработку данных в соответствии с вариантом задания. Исходные данные должны вводиться посредством формы ввода, в которой используются поля типа TEdit и TComboBox. Результат ввода отображается в таблице типа TStringGrid. Промежуточные результаты выборки могут быть выведены в таблицу типа TStringGrid или в текстовое поле типа TMemo. Итоговый результат расчётов и вычислений может быть выведен в виде текстового ответа TLabel или добавлен в текстовое поле типа TMemo.</p> <p>Варианты:</p> <ol style="list-style-type: none"> <li>1. Создать запись «Покупатель» с полями: «Имя» (строка), «Пол» (символ),</li> </ol>	<p>ПК-2: ИД-ПК-2.3 ИД-ПК-2.4</p>

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		<p>«Цена товара» (вещественное число), «Количество товаров» (целое число). Имена покупателей могут повторяться, т.е. один покупатель может покупать разные товары в разном количестве. Вывести информацию о покупателе-мужчине, купившем больше всего товаров.</p> <p>2. Создать запись «Покупатель» с полями: «Имя» (строка), «Пол» (символ), «Цена товара» (вещественное число), «Количество товаров» (целое число). Имена покупателей могут повторяться, т.е. один покупатель может покупать разные товары в разном количестве. Вывести информацию о покупателе-женщине, потратившей больше всего денег.</p> <p>3. Создать запись «Покупатель» с полями: «Имя» (строка), «Пол» (символ), «Цена товара» (вещественное число), «Количество товаров» (целое число). Имена покупателей могут повторяться, т.е. один покупатель может покупать разные товары в разном количестве. Вывести информацию о покупателе-женщине, купившей больше всего товаров.</p> <p>4. Создать запись «Покупатель» с полями: «Имя» (строка), «Пол» (символ), «Цена товара» (вещественное число), «Количество товаров» (целое число). Имена покупателей могут повторяться, т.е. один покупатель может покупать разные товары в разном количестве. Вывести информацию о покупателе-мужчине, потратившем больше всего денег.</p> <p>5. Создать запись «Покупатель» с полями: «Имя» (строка), «Пол» (символ), «Цена товара» (вещественное число), «Количество товаров» (целое число). Имена покупателей могут повторяться, т.е. один покупатель может покупать разные товары в разном количестве. Вывести информацию о покупателе-мужчине, потратившем меньше всего денег.</p> <p>...</p>	
	<p>Выполнение лабораторной работы. Лабораторная работа №</p>	<p>Использование подпрограмм и модулей. Создать приложение с графическим интерфейсом и двумя визуальными формами, производящее обработку данных одномерного массива в соответствии</p>	<p>ПК-2: ИД-ПК-2.1 ИД-ПК-2.2</p>

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
	3.1	<p>с вариантом задания.</p> <p>Значения элементов массива должны вводиться в одной форме, а все операции вывода производиться во второй форме. Данные должны обрабатываться в отдельно созданной подпрограмме (процедуре или функции согласно варианту задания). Массив и количество элементов массива должны передаваться в подпрограмму в качестве параметров.</p> <p>Исходные данные должны вводиться посредством формы ввода, в которой используется поле типа TEdit. Исходный массив должен отображаться в текстовом поле типа TMemo или в таблице типа TStringGrid. Если в результате вычислений элементы массива изменяются, то полученный массив также необходимо вывести в текстовом поле типа TMemo или в таблице типа TStringGrid. Итоговый результат расчётов и вычислений может быть выведен в виде текстового ответа TLabel или добавлен в текстовое поле типа TMemo.</p> <p>Варианты:</p> <ol style="list-style-type: none"> <li>1. Массив целых чисел. В процедуре отсортировать массив по возрастанию.</li> <li>2. Массив целых чисел. В процедуре отсортировать массив по убыванию.</li> <li>3. Массив вещественных чисел. В процедуре отсортировать массив по возрастанию.</li> <li>4. Массив вещественных чисел. В процедуре отсортировать массив по убыванию.</li> <li>5. Массив целых чисел. В процедуре удалить из массива четные элементы.</li> </ol> <p>Размерность итогового массива должна уменьшиться.</p> <p>...</p>	ИД-ПК-2.3 ИД-ПК-2.4
	Выполнение лабораторной работы. Лабораторная работа № 3.2	<p>Использование библиотек DLL.</p> <p>Создать приложение с графическим интерфейсом, производящее вычисления и обработку данных в подпрограммах, находящихся в библиотеках DLL, в соответствии с вариантом задания. Исходные данные должны вводиться в полях типа TEdit. Результаты расчётов и вычислений могут быть выведены в текстовое поле типа TMemo.</p> <p>Исходными данными являются два целых числа. С ними необходимо произвести вычисления в соответствии с вариантом задания.</p> <p>Вычисления должны быть произведены в подпрограммах. Подпрограммы должны находиться в библиотеках DLL:</p> <ol style="list-style-type: none"> <li>1) Функция: результат вычислений должен быть получен в качестве</li> </ol>	ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3 ИД-ПК-2.4

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		<p>возвращаемого значения функции. <b>Использовать статический вызов.</b>            2) Процедура: результат вычислений должен быть получен в виде параметра-переменной. <b>Использовать динамический вызов.</b>            Ввод данных и вывод результатов должны производиться в главной визуальной форме, все вычисления – в подпрограммах.            Варианты:      1) функции                              2) процедуры</p> <p>1. 1) <math>X = \begin{cases} a * b + 1, &amp; \text{если } a &gt; b, \\ 25, &amp; \text{если } a = b, \\ (a - 5) / b, &amp; \text{если } a &lt; b; \end{cases}</math>      2) <math>X = \begin{cases} a * b - 3, &amp; \text{если } a &gt; b, \\ 2, &amp; \text{если } a = b, \\ (a^3 + 1) / b, &amp; \text{если } a &lt; b; \end{cases}</math></p> <p>2. 1) <math>X = \begin{cases} a / b + 5, &amp; \text{если } a &lt; b, \\ -5, &amp; \text{если } a = b, \\ (a * a - b) / b, &amp; \text{если } a &gt; b; \end{cases}</math>      2) <math>X = \begin{cases} a / b + 1, &amp; \text{если } a &lt; b, \\ -1, &amp; \text{если } a = b, \\ (a * b - 5) / a, &amp; \text{если } a &gt; b; \end{cases}</math></p> <p>3. 1) <math>X = \begin{cases} a / b - 1, &amp; \text{если } a &gt; b, \\ -25, &amp; \text{если } a = b, \\ (a^3 - 5) / a, &amp; \text{если } a &lt; b; \end{cases}</math>      2) <math>X = \begin{cases} a * b + 21, &amp; \text{если } a &gt; b, \\ -5, &amp; \text{если } a = b, \\ 3 * a / b + 1, &amp; \text{если } a &lt; b; \end{cases}</math></p> <p>4. 1) <math>X = \begin{cases} 5 * a + b, &amp; \text{если } a &gt; b, \\ -125, &amp; \text{если } a = b, \\ (a - 5) / b, &amp; \text{если } a &lt; b; \end{cases}</math>      2) <math>X = \begin{cases} a * b - 1, &amp; \text{если } a &gt; b, \\ 255, &amp; \text{если } a = b, \\ (a - 5) / b, &amp; \text{если } a &lt; b; \end{cases}</math></p> <p>5. 1) <math>X = \begin{cases} b * a + 1, &amp; \text{если } a &gt; b, \\ -10, &amp; \text{если } a = b, \\ (a - 5) / b, &amp; \text{если } a &lt; b; \end{cases}</math>      2) <math>X = \begin{cases} a / b + 31, &amp; \text{если } a &gt; b, \\ -25, &amp; \text{если } a = b, \\ (a * 5 - 1) / a, &amp; \text{если } a &lt; b; \end{cases}</math></p> <p>...</p>	
	Выполнение лабораторной работы. Лабораторная работа №	Поиск файлов. Работа с найденными текстовыми файлами. Создать приложение с графическим интерфейсом, производящее поиск и обработку в соответствии с вариантом задания всех текстовых файлов с заданным именем. При каждом	ПК-2: ИД-ПК-2.1 ИД-ПК-2.2



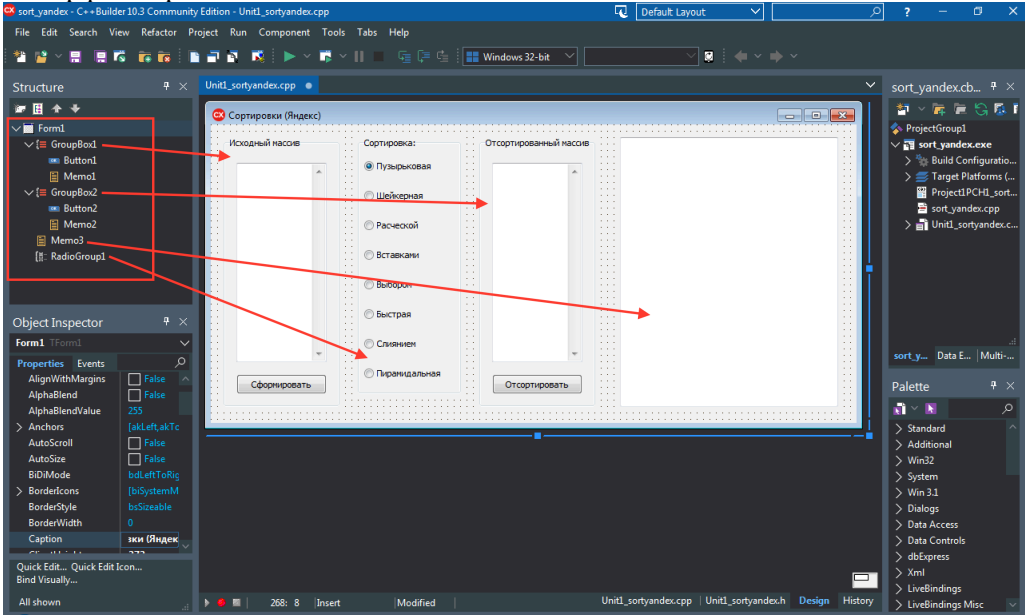
№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
	4.1	<p>новом поиске надо создавать текстовый файл, в который будут записываться все результаты выполнения приложения, то есть необходимо продублировать на экран и в текстовый файл весь вывод информации о просмотренных папках, о содержимом каждого текстового файла и о результатах вычислений. Этот файл перезаписывается при каждом запуске приложения, и в нём при завершении работы приложения можно просмотреть всю информацию. Приложение должно состоять из двух форм. Первая форма содержит элементы ввода, вывода и управления, связанные с поиском файлов. Вторая форма содержит элементы ввода, вывода и управления, связанные с обработкой каждого найденного файла. Исходные данные для поиска вводятся в полях типа TEdit. Результаты поиска и вывода содержимого каждого файла могут производиться в текстовое поле типа TMemo. Результаты вычислений по каждому найденному файлу могут добавляться в текстовое поле типа TMemo, или выводиться в текстовые сообщения типа TLabel, или выводиться в информационном окне. Поиск файлов производится во всех вложенных папках (каталогах/директориях), начиная с указанной, все проверяемые папки должны выводиться на экран и записываться в текстовый файл. Одноименные файлы могут находиться в разных папках, необходимо поочередно выводить на экран и записывать в текстовый файл их содержимое. Если ни одного заданного файла не найдено, необходимо вывести соответствующее сообщение. Для проверки работы приложения целесообразно в разных вложенных папках создать несколько файлов с одинаковыми именами и с текстами, соответствующими вариантам заданий.</p> <p>Варианты:</p> <ol style="list-style-type: none"> <li>1. Подсчитать количество строк в файле и количество символов «;» во всём файле.</li> <li>2. Определить максимальную длину строки в файле, подсчитать количество символов «.» во всём файле.</li> <li>3. Определить минимальную длину строки в файле, подсчитать количество символов «0» (ноль) во всём файле.</li> <li>4. Определить, присутствует ли в файле сочетание символов «unit» (не зависящее от регистра символов), подсчитать количество и вывести номера строк, в которых оно обнаружено.</li> <li>5. Определить, присутствует ли в файле сочетание символов «for» (не зависящее</li> </ol>	ИД-ПК-2.3 ИД-ПК-2.4

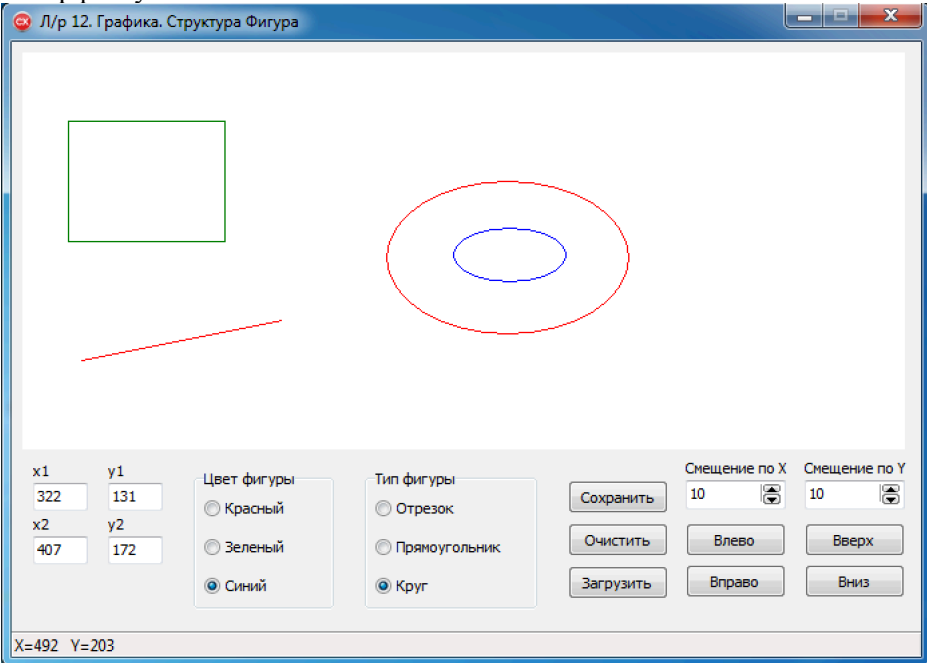
№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		от регистра символов), подсчитать количество и вывести номера строк, в которых оно обнаружено. ...	
	Выполнение лабораторной работы. Лабораторная работа № 4.2	<p>Обработка данных в типизированных файлах. Создать приложение с графическим интерфейсом для создания двух типизированных файлов, заполняемых случайным набором целочисленных значений, производящее обработку данных в этих файлах в соответствии с вариантом задания и записывающее результат в новый типизированный файл. Исходные данные и результаты должны отображаться в текстовых полях типа TMemo или в таблицах типа TStringGrid. Массивы и множества не использовать.</p> <p>Варианты:</p> <ol style="list-style-type: none"> <li>1. Заполнить 2 исходных файла случайным количеством случайных значений. Найти минимальное значение в 1-м файле. Из 1-го файла в результирующий файл записать данные до первого значения минимума. Из 2-го файла записать только значения больше минимума.</li> <li>2. Заполнить 2 исходных файла случайным количеством случайных значений. Найти максимальное значение во 2-м файле. Из 1-го файла в результирующий файл записать только значения меньше максимума. Из 2-го файла в результирующий файл записать данные до первого значения максимума.</li> <li>3. Заполнить 2 исходных файла случайным количеством случайных значений. Найти максимальное значение в 1-м файле и минимальное во 2-м файле. Из 2-го файла в результирующий файл записать только значения больше максимума. Из 1-го файла в результирующий файл записать только значения меньше минимума.</li> <li>4. Заполнить 2 исходных файла случайным количеством случайных значений. Найти минимальное значение в 1-м файле и максимальное во 2-м файле. Из 1-го файла в результирующий файл записать только значения больше максимума. Из 2-го файла в результирующий файл записать только значения меньше минимума.</li> <li>5. Заполнить 2 исходных файла случайным количеством случайных значений. Найти минимальное значение в 1-м файле и максимальное во 2-м файле. Из 1-го файла в результирующий файл записать данные до первого значения минимума. Из 2-го файла в результирующий файл записать данные после первого значения максимума.</li> </ol> <p>...</p>	ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3 ИД-ПК-2.4

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
	Выполнение лабораторной работы. Лабораторная работа № 5.1	<p>Статическая графика в окнах Windows.</p> <p>Создать приложение с графическим интерфейсом, формирующее векторное изображение в соответствии с вариантом задания. Для вывода графики использовать визуальные компоненты TPaintBox или TImage, рисование производить с использованием операторов ObjectPascal, свойств и методов визуальных компонентов среды разработки приложений Lazarus и GDI API операционной системы.</p> <p>Все элементы изображения должны иметь относительные координаты, всё изображение в целом должно без искажений масштабироваться в зависимости от изменения размеров рабочей формы приложения.</p> <p>При создании элементов изображения обязательно использовать:</p> <ul style="list-style-type: none"> <li>- различные типы заливки, определяемые свойствами визуальных компонентов;</li> <li>- различные типы заливки, загружаемые из внешних источников-образцов;</li> <li>- циклические конструкции для повторяющихся элементов изображения;</li> <li>- многоточечные фигуры-полигоны на основе массивов;</li> <li>- текстовые надписи с настройками шрифтов.</li> </ul> <p>Варианты:</p> <ol style="list-style-type: none"> <li>1. Стоящий на столе компьютер.</li> <li>2. Одноэтажный дом с крышей, окнами, дверью и лестницей.</li> <li>3. Магазин с витриной и вывеской.</li> <li>4. Пилотируемый воздушный шар с корзиной на фоне неба.</li> <li>5. Телефон с кнопочным набором.</li> </ol> <p>...</p>	ПК-2: ИД-ПК-2.3 ИД-ПК-2.4
	Выполнение лабораторной работы. Лабораторная работа № 5.2	<p>Динамическая графика в окнах Windows.</p> <p>Создать приложение с графическим интерфейсом, обрабатывающее нажатия на левую кнопку «мыши» в соответствии с вариантом задания и обеспечивающее перерисовку созданного изображения в зависимости от изменения размеров окна. Для вывода графики использовать визуальные компоненты TPaintBox или TImage, рисование производить с использованием операторов ObjectPascal, свойств и методов визуальных компонентов среды разработки приложений Lazarus и GDI API операционной системы.</p> <p>Графическую область вывода необходимо разделить на заданное в варианте количество частей и выводить в каждую часть элементы изображения в</p>	ПК-2: ИД-ПК-2.3 ИД-ПК-2.4

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		<p>соответствии с вариантом задания. Разделяющие область отрезки должны изменяться (масштабироваться) в соответствии с изменениями размеров формы приложения. Геометрические фигуры должны иметь центр в точке нажатия «мыши», текстовые сообщения должны выводиться, начиная от верхней левой точки нажатия «мыши».</p> <p>Все элементы изображения должны иметь относительные координаты. При перерисовке графической области (например, во время изменения размеров формы приложения), созданные с помощью «мыши» фигуры и текстовые надписи должны сохранять исходные цвета, стили заливки и настройки шрифта, а их местоположение должно масштабироваться в зависимости от изменения размеров графической области приложения.</p> <p>Варианты:</p> <p><b>1-11.</b> Разделить графическую область двумя отрезками: вертикальным и горизонтальным, проходящими по центру. Должны получиться 4 сектора графической области.</p> <p>1. Если кнопка «мыши» нажата в Секторах 1 и 3, то выводить квадраты разными стилями заливки для каждого сектора, а в Секторах 2 и 4 выводить текст «Сектор &lt;N&gt;» разными типами шрифта для каждого сектора и с указанием соответствующего номера &lt;N&gt;.</p> <p>2. Если кнопка «мыши» нажата в Секторах 1 и 3, то выводить круги разными стилями заливки для каждого сектора, а в Секторах 2 и 4 выводить текст «Сектор &lt;N&gt;» разными типами шрифта для каждого сектора и с указанием соответствующего номера &lt;N&gt;.</p> <p>3. Если кнопка «мыши» нажата в Секторах 1 и 3, то выводить круги разными цветами сплошной заливки для каждого сектора, а в Секторах 2 и 4 выводить квадраты разными стилями заливки для каждого сектора.</p> <p>4. Если кнопка «мыши» нажата в Секторах 1 и 3, то выводить квадраты разными стилями заливки для каждого сектора, а в Секторах 2 и 4 выводить круги разными цветами сплошной заливки для каждого сектора.</p> <p>5. Если кнопка «мыши» нажата в Секторах 1 и 3, то выводить текст «Сектор &lt;N&gt;» разными типами шрифта для каждого сектора и с указанием соответствующего номера &lt;N&gt;, а в Секторах 2 и 4 выводить круги разными цветами сплошной заливки для каждого сектора.</p>	

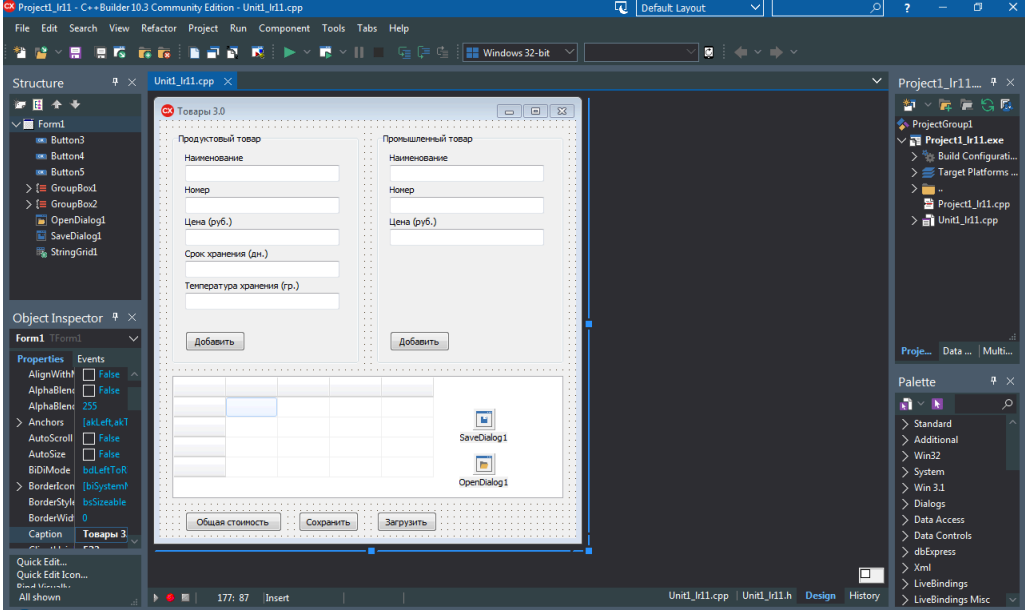
№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
	Выполнение лабораторной работы. Лабораторная работа № 6.1	<p>Использование указателей.</p> <p>Создать приложение с графическим интерфейсом для вычислений в соответствии с вариантом задания. Все данные должны иметь динамическую типизацию, т.е. объявляться и использоваться посредством указателей. В приложении должна быть создана процедура для вычислений, параметры в которую также передаются как указатели. Значения исходных данных для вычислений должны вводиться в редактируемых элементах типа TEdit.Text.</p> <p>Варианты:</p> <ol style="list-style-type: none"> <li>Использовать указатель на целый тип Int64 и указатель на вещественный тип Real48. Вычислить результаты: 1) вещественного деления <math>\langle \text{значение вещественного типа} \rangle / \langle \text{значение вещественного типа} \rangle</math> на целый тип; 2) целочисленного деления <math>\langle \text{значение целого типа} \rangle / \langle \text{значение целого типа} \rangle</math> целой части вещественного типа на целый тип.</li> <li>Использовать указатель на целый тип Longint и указатель на вещественный тип Double. Вычислить результаты: 1) вещественного деления <math>\langle \text{значение вещественного типа} \rangle / \langle \text{значение вещественного типа} \rangle</math> целого типа на вещественный тип; 2) целочисленного деления <math>\langle \text{значение целого типа} \rangle / \langle \text{значение целого типа} \rangle</math> целого типа на целую часть вещественного типа.</li> <li>Использовать указатель на целый тип Word и указатель на вещественный тип Single. Поменять местами значение целого типа и значение целой части вещественного типа.</li> <li>Использовать указатель на целый тип Integer и указатель на вещественный тип Extended. Получить результат вычислений по формуле: <math>\langle \text{значение целого типа} \rangle / 2 + \langle \text{значение вещественного типа} \rangle - 2</math>.</li> <li>Использовать указатель на целый тип Cardinal и указатель на вещественный тип Double. Получить результат вычислений по формуле: <math>\langle \text{значение дробной части вещественного типа} \rangle * 100 - \langle \text{значение целого типа} \rangle</math>.</li> </ol> <p>...</p>	ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3 ИД-ПК-2.4
	Выполнение лабораторной работы. Лабораторная работа № 7.1	<p>Сортировки Яндекс.</p> <p>Лабораторная работа посвящена изучению основных алгоритмов сортировок и способов их реализации на примере Яндекс (см. <a href="https://academy.yandex.ru/posts/osnovnye-vidy-sortirovok-i-primery-ikh-realizatsii">https://academy.yandex.ru/posts/osnovnye-vidy-sortirovok-i-primery-ikh-realizatsii</a>).</p> <p>Лабораторную работу необходимо реализовать в виде визуального приложения. Нужно изучить представленный иллюстрированный пример реализации на C++</p>	ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3 ИД-ПК-2.4

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		<p>основных видов сортировок.</p> <p>Пример реализации на С++ основных видов сортировок:</p> <ol style="list-style-type: none"> <li>1) Кнопка 1 формирует случайным образом массив целых чисел из 20000 элементов и выводит, например, в MEMO1.</li> <li>2) НА ФОРМЕ ИМЕЕТСЯ радиогрупп (8 радиоточек для каждой из сортировок).</li> <li>3) Кнопка 2 - сортирует массив случайных чисел (вызывается функция для выбранной пользователем сортировки).</li> <li>4) Выводим отсортированный массив и время в миллисекундах, затраченное в ходе сортировки.</li> </ol> <p>Интерфейс приложения:</p>  <p>Необходимо воспроизвести реализацию проекта и по аналогии сделать пирамидальную сортировку и сортировку слиянием. Выполнить оптимизацию кода (для ускорения операций) и абстракцию.</p> <p>Визуализация сортировки Яндекса.</p> <p>Лабораторная работа посвящена визуализации алгоритмов сортировки Яндекса</p>	

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		<p>(см. <a href="https://academy.yandex.ru/posts/osnovnye-vidy-sortirovok-i-primery-ikh-realizatsii">https://academy.yandex.ru/posts/osnovnye-vidy-sortirovok-i-primery-ikh-realizatsii</a>). Лабораторную работу необходимо выполнить в виде визуального приложения. Разрабатывается визуальное приложение, позволяющее визуализировать процесс сортировки целочисленного массива из 12 случайных элементов с анимацией. Допускается использовать подходящий стандартный компонент, либо конструкцию из прямоугольников. Визуализируемую сортировку пользователь выбирает в списке (компонент ComboBox). Замедление просто реализовать с помощью связки <code>sleep()</code> + <code>ProcessMessages()</code>.</p>	
	<p>Выполнение лабораторной работы. Лабораторная работа № 7.2</p>	<p>Лабораторная работа посвящена изучению структур и ДСД. Реализовать взаимодействия с геометрическими примитивами и отработать структуры. Создать визуальное приложение согласно представленному интерфейсу.</p>  <p>Создать приложение, демонстрирующее работу списков, очередей и стеков. 1)</p>	<p>ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3 ИД-ПК-2.4</p>

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		<p>Организовать считывание текстового файла с помощью структуры стек. 2) В л/р про авиарейсы нужно добавить поле дата вылета. Соответственно, при добавлении рейса необходимо формировать структуру очередь, согласно датам вылета самолетов. Нужно создать кнопку "Очередность вылетов", которая выведет созданную структуру. Создать визуальное приложение согласно представленному интерфейсу.</p> <p>Написать программу учёта для автосервиса, выполняющего кузовные работы и ремонт двигателей. При записи на обслуживание заполняется заявка, в которой указываются фамилия владельца, марка автомобиля, вид работы, дата приёма заказа и стоимость ремонта. После выполнения работы распечатывается квитанция (сохранение заявки в файл). Для хранения информации использовать динамическую структуру данных – очередь. Интерфейс программы организовать следующим образом: 1) добавление заявки; 2) распечатка квитанции (сохранение в файл); 3) вывод списка заявок на экран; 4) запись списка заявок в файлы (1 – кузовные работы; 2 – ремонт двигателя); 5) выход.</p> <p>Программу написать с использованием динамической структуры данных "Очередь" и компонента StringGrid.</p>	
	<p>Выполнение лабораторной работы. Лабораторная работа № 7.3</p>	<p>Классы. Наследование и виртуальные методы.</p> <p>Лабораторная работа посвящена изучению классов в ООП на C++. Необходимо изучить и повторить пример, демонстрирующий применение технологии наследования с виртуальными методами. Взяв за основу проект из предыдущей лабораторной работы, мы создадим его улучшенную версию, которая позволит работать не с двумя массивами, а с одним универсальным, который может хранить данные как продуктовых товаров, так и промышленных. Также реализуем сохранение данных в текстовый файл и считывание их из него в компонент StringGrid (по аналогии с предыдущей л/р). Также реализуем нормальное функционирование кнопки "Общая стоимость", которая сможет рассчитывать общую стоимость как добавленных вручную товаров, так и загруженных из файла.</p> <p>Разместим на форме необходимые для UI компоненты как на скриншоте ниже.</p>	<p>ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3 ИД-ПК-2.4</p>



№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		 <p>В .h-файле пропишем классы (базовый и производные). Методы базового класса объявим виртуальными. Выдержка из лекции: "Виртуальные функции позволяют решать прямо в процессе выполнения программы, какую именно функцию вызывать. Виртуальные функции дают большую гибкость при выполнении одинаковых действий над разнородными объектами. В частности, они разрешают использование функций, вызванных из массива указателей на базовый класс, который на самом деле содержит указатели (или ссылки) на множество порождённых классов. Это пример полиморфизма."</p> <p>Переключимся на unit.cpp и распишем все методы и глобальные переменные, как в программном коде ниже.</p> <pre>int VidTovara = 0; // 0-продуктовые, 1-промышленные int i=0, ij=0; // глобальные переменные счетчики void Tovar::dobav_zap() { // метод для добавления записей     if(VidTovara==0) {</pre>	

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		<pre> strcpy(name, AnsiString(Form1-&gt;LabeledEdit1-&gt;Text).c_str()); number = StrToInt(Form1-&gt;LabeledEdit2-&gt;Text); price = StrToFloat(Form1-&gt;LabeledEdit3-&gt;Text); } if(VidTovara==1) { strcpy(name, AnsiString(Form1-&gt;LabeledEdit6-&gt;Text).c_str()); number = StrToInt(Form1-&gt;LabeledEdit7-&gt;Text); price = StrToFloat(Form1-&gt;LabeledEdit8-&gt;Text); } } void Товар::dobav_zap_iz_SG() { strcpy(name, AnsiString(Form1-&gt;StringGrid1-&gt;Cells[0][ij]).c_str()); number = StrToInt(Form1-&gt;StringGrid1-&gt;Cells[1][ij]); price = StrToFloat(Form1-&gt;StringGrid1-&gt;Cells[2][ij]); } void Товар::show() { // метод для показа записей в стринггриде Form1-&gt;StringGrid1-&gt;Cells[0][i] = AnsiString(name); Form1-&gt;StringGrid1-&gt;Cells[1][i] = number; Form1-&gt;StringGrid1-&gt;Cells[2][i] = price; } float Товар::get_price() { // чтобы не использовать прямой доступ к полю price return price; } void Товар::clear() { // метод очистки эдитов if(VidTovara == 0) { Form1-&gt;LabeledEdit1-&gt;Clear(); Form1-&gt;LabeledEdit2-&gt;Clear(); Form1-&gt;LabeledEdit3-&gt;Clear(); } if(VidTovara == 1) { Form1-&gt;LabeledEdit6-&gt;Clear(); Form1-&gt;LabeledEdit7-&gt;Clear(); Form1-&gt;LabeledEdit8-&gt;Clear(); } } </pre>	

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		<pre> } }  void ТоварProd::dobav_zap() {     Товар::dobav_zap(); // вызов метода базового класса для первых трех полей     srok = StrToInt(Form1-&gt;LabeledEdit4-&gt;Text);     temp = StrToInt(Form1-&gt;LabeledEdit5-&gt;Text); }  void ТоварProd::dobav_zap_iz_SG() {     Товар::dobav_zap_iz_SG(); // вызов метода базового класса     srok = StrToInt(Form1-&gt;StringGrid1-&gt;Cells[3][ij]);     temp = StrToInt(Form1-&gt;StringGrid1-&gt;Cells[4][ij]); }  void ТоварProd::show() {     Товар::show(); // вызов метода базового класса для первых трех полей     Form1-&gt;StringGrid1-&gt;Cells[3][i] = srok;     Form1-&gt;StringGrid1-&gt;Cells[4][i] = temp; }  void ТоварProd::clear() {     Товар::clear();     Form1-&gt;LabeledEdit4-&gt;Clear();     Form1-&gt;LabeledEdit5-&gt;Clear(); }  const int K = 100; // максимальное число товаров Товар *prod_and_prom[K]; // !!! массив может хранить продтовары и промтовары !!! - одно из главных достоинств виртуальных методов Товар *new_prom_and_prod[K]; // новый массив пром и прод товаров (для стрингрида, заполненного из файла)  Распишите обработчики добавления товаров. Далее, загружаем данные из файла в список StringList, из которого заполняем StringGrid. Разбираем считываемые данные на принадлежность к промышленным или продуктовым товарам и </pre>	

№ пп	Формы текущего контроля	Примеры типовых заданий	Формируемая компетенция
		заполняем новый массив, который создали предварительно именно для этих целей. Реализовав рассмотренный пример, необходимо добавить возможности: а) сортировки товаров по наименованию и цене; б) поиска (по всем свойствам товаров).	
2	Посещение профориентационных мероприятий	№1. Участие в публичных профориентационных мероприятиях, проводимых на территории РГУ им. А.Н. Косыгина. №2. Участие в публичных профориентационных мероприятиях, проводимых вне территории РГУ им. А.Н. Косыгина.	ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3 ИД-ПК-2.4
3	Участие (достижения) в профессиональных конкурсах	Участие или призовое место в хакатоне или ином соревновании с официальным участием РГУ им. А.Н. Косыгина	
4	Научная и/или практическая работа	Участие в научной конференции или ином научном мероприятии в качестве представителя РГУ им. А.Н. Косыгина	

## 5.2. Критерии, шкалы оценивания текущего контроля успеваемости:

Критерии и шкалы оценивания формируются в соответствии с ограничениями Методикой использования балльно-рейтинговой системы при реализации основных профессиональных образовательных программ высшего образования Института информационных технологий и цифровой трансформации.

Тип контрольно-рейтингового мероприятия	Наименование КРМ	Критерии оценивания и правила начисления баллов за КРМ			Балл или диапазон баллов
		Контрольные сроки и шкала эрозии баллов	Правила начисления баллов	Начисление баллов после завершения аттестации	

Тип контрольно-рейтингового мероприятия	Наименование КРМ	Критерии оценивания и правила начисления баллов за КРМ			Балл или диапазон баллов
		Контрольные сроки и шкала эрозии баллов	Правила начисления баллов	Начисление баллов после завершения аттестации	
Посещение проф-ориентационных мероприятий	Участие в публичных мероприятиях, проводимых на территории РГУ им. А.Н. Косыгина	Нет	Приказ или Распоряжение о включении мероприятий в учебный процесс, наличие отметки о посещении мероприятия. Подтверждение от директора института о соответствии мероприятия профилю подготовки. Балл за КРМ определяется как отношение количества посещенных мероприятий к проведенным. Мероприятие засчитывается как посещенное при условии активной работы обучающегося на мероприятии: озвучивание вопросов, участие в дискуссиях, проявлении признаков сформированности соответствующих компетенций и т.п. КРМ может быть учтено по всем дисциплинам, использующим БРС.	Нет	1-5
	Участие в публичных мероприятиях, проводимых вне территории РГУ им. А.Н. Косыгина	Нет	Приказ или Распоряжение об участии в мероприятии, наличие подтверждения посещения мероприятия. Подтверждение от директора института о соответствии мероприятия профилю подготовки. Балл за КРМ определяется как отношение количества посещенных мероприятий к проведенным. Мероприятие засчитывается как посещенное при условии активной работы обучающегося на мероприятии: озвучивание вопросов, участие в дискуссиях, проявлении признаков сформированности соответствующих компетенций и т.п. КРМ может быть учтено по всем дисциплинам, использующим БРС.	Нет	1-4
Участие (достижения) в профессиональных конкурсах	Участие или призовое место в хакатоне или ином соревновании с официальным участием РГУ им. А.Н. Косыгина	Нет	Приказ или Распоряжение об организации и/или участии в мероприятии. Документы, подтверждающие участие и результаты участия. Соответствие содержания дисциплины и мероприятия определяет реализующий дисциплину преподаватель. Баллы за мероприятия определяются реализующим дисциплину преподавателем на основании предоставленных документов. КРМ может быть учтено только в одной дисциплине, использующей БРС (по выбору студента).		
			Обучающийся проявил профессиональный подход к выполнению конкурсного задания, занял призовое место или его конкурсная работа выполнена на высоком профессиональном уровне без грубых ошибок.	Да	1-2

Тип контрольно-рейтингового мероприятия	Наименование КРМ	Критерии оценивания и правила начисления баллов за КРМ			Балл или диапазон баллов
		Контрольные сроки и шкала эрозии баллов	Правила начисления баллов	Начисление баллов после завершения аттестации	
			Обучающийся участвовал в конкурсе, выполнил конкурсное задание полностью и в срок. Однако его работа содержит ошибки, пометки или не соответствует тематике дисциплины.		0-1
Научная и/или практическая работа	Участие в научной конференции или ином научном мероприятии в качестве представителя РГУ им. А.Н. Косыгина	Нет	Сертификат или иные документ, подтверждающие участие и результаты участия в научных конференциях или иных научных мероприятиях. Соответствие содержания дисциплины и прошедшего обучения определяет реализующий дисциплину преподаватель. Баллы за мероприятия определяются реализующим дисциплину преподавателем на основании предоставленных документов.  КРМ может быть учтено только в одной дисциплине, использующей БРС (по выбору студента).		
			Обучающийся представил актуальную и оригинальную работу, соответствующую тематике дисциплины. Работа отмечена призовым местом, иным знаком отличия или представляет собой интерес в рамках ИТ-направления.	Да	3-4
			Обучающийся представил формальную работу, не имеющей признаки научной работы. Работа содержит ошибки, признаки плагиата или не соответствует научной тематике по формальным признакам.		0-2
Выполнение учебных заданий	Лабораторная работа	Нет	Работа выполнена полностью. Нет ошибок в логических рассуждениях и в реализации задания в виде файла или выполняемой программы. Возможно наличие одной неточности или описки, не являющиеся следствием незнания или непонимания учебного материала и не влияющей на функциональные качества программы. Обучающийся показал полный объем знаний, умений в освоении пройденных тем и применение их на практике. Работа зачтена.	Да	47-55
			Работа выполнена полностью, но выбран неэффективный алгоритм или метод реализации, обоснований шагов решения недостаточно. Допущена одна ошибка или два-три недочета, которые незначительно влияют на качество представленной работы. Работа зачтена.		38-46

Тип контрольно-рейтингового мероприятия	Наименование КРМ	Критерии оценивания и правила начисления баллов за КРМ			Балл или диапазон баллов
		Контрольные сроки и шкала эрозии баллов	Правила начисления баллов	Начисление баллов после завершения аттестации	
			Допущены более одной ошибки или более двух-трех недочетов, которые оказывают значительное влияние на представляемый файл или компьютерную программу, ухудшают их информативность и функциональные возможности. Работа зачтена.		30-37
			Работа выполнена не полностью. Допущены грубые ошибки. Файлы не содержат необходимой информации, компьютерная программа выдаёт неправильные результаты при вычислении тестовых примеров. Работа не зачтена.		0-29
Итого:					0-70

### 5.3. Промежуточная аттестация:

Форма промежуточной аттестации	Типовые контрольные задания и иные материалы для проведения промежуточной аттестации:	Формируемая компетенция
Экзамен: Компьютерное тестирование	<p>1. Какой из типов данных относится к скалярным?</p> <p>а) Объектный б) Вещественный в) Строковый г) Массив</p> <p>2. Какой из типов данных относится к структурированным?</p> <p>а) Целый б) Вещественный в) Строковый г) Указатель</p> <p>...</p> <p>12. Параметры, используемые при вызове подпрограмм и имеющие конкретные значения, называются...</p> <p>а) формальными</p>	ПК-2: ИД-ПК-2.1 ИД-ПК-2.2 ИД-ПК-2.3 ИД-ПК-2.4

	<p>b) фактическими  c) вариативными  d) стохастическими</p> <p>13. Какое из определений не относится к файловому типу данных в языке высокого уровня?  a) У файла есть имя-идентификатор  b) При работе с файлами могут выполняться операции ввода - вывода  c) Количество компонент в файле задаётся заранее и не может меняться  d) Размер файла имеет переменное значение и ограничивается только физическими ресурсами компьютера</p> <p>...</p> <p>17. Как с помощью логических операций записать, что значение x находится в интервале от 0 до 20?  a) <math>(x \geq 0)</math> И <math>(x \leq 20)</math>  b) <math>(x \geq 0)</math> ИЛИ <math>(x \leq 20)</math>  c) <math>(x &lt; 0)</math> ИЛИ <math>(x &lt; 20)</math>  d) <math>(x &lt; 0)</math> И <math>(x &gt; 20)</math></p> <p>18. В результате выполнения следующей программы</p> <pre>----- X присвоить 6; Y присвоить 10; if (x &lt; 2) x присвоить -3; y присвоить 15; -----</pre> <p>переменные станут равны</p> <p>a) <math>x = 6, y = 10</math>  b) <math>x = 6, y = 15</math>  c) <math>x = -3, y = 10</math>  d) <math>x = -3, y = 15</math></p> <p>...</p>	
--	---	--

#### 5.4. Критерии, шкалы оценивания промежуточной аттестации учебной дисциплины:

Результат промежуточной аттестации определяется как соответствие суммы набранных рейтинговых баллов за контрольно-рейтинговые мероприятия текущей аттестации и контрольно-рейтинговых баллов, набранных за промежуточную аттестацию. Оценка по дисциплины выставляется в



соответствии с Системой оценивания результатов текущего контроля и промежуточной аттестации, описанной в данном документе, а также в соответствии с Методикой использования балльно-рейтинговой системы при реализации основных профессиональных образовательных программ высшего образования Института информационных технологий и цифровой трансформации.

Форма промежуточной аттестации	Критерии оценивания	Шкалы оценивания		
Наименование оценочного средства		100-балльная система	Пятибалльная система	
Экзамен: компьютерное тестирование	<p>За выполнение каждого тестового задания испытуемому выставляются баллы. За полностью правильный ответ к каждому заданию с выбором одного правильного варианта выставляется один балл, за неправильный — ноль. За задания с выбором нескольких правильных ответов или в заданиях с сопоставлениями испытуемый может получить менее 1 балла. Например, если правильных ответов в задании два, то за каждый он получает 0,5 балла, если правильных ответов три, то за каждый он получает 0,333 балла и т.п.</p> <p>Правила оценки всего теста: вне зависимости от количества заданий в тесте общая сумма баллов за все правильные ответы пересчитывается тестирующей компьютерной системой в итоговые баллы. 20 итоговых баллов эквивалентны 100% правильных ответов. Для того, чтобы получить отличную, хорошую, удовлетворительную или неудовлетворительную оценки, итоговые баллы за промежуточную аттестацию складываются с баллами за выполненные лабораторные работы и практические задания.</p>	21 – 30 баллов	5	85% - 100%
		11 – 20 баллов	4	70% - 84%
		6 – 10 баллов	3	55% - 69%
		0 – 5 баллов	2	54% и менее 54%

### 5.5. Система оценивания результатов текущего контроля и промежуточной аттестации.

В соответствии с Методикой использования балльно-рейтинговой системы при реализации основных профессиональных образовательных программ высшего образования Института информационных технологий и цифровой трансформации, оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.

Форма контроля	100-балльная система	Пятибалльная система
<b>Текущий контроль:</b>		
- выполнение лабораторных работ	0 – 55 баллов	зачтено/не зачтено
- посещение профориентационных мероприятий	0 – 9 баллов	зачтено/не зачтено
- участие (достижения) в профессиональных конкурсах	0 – 3 балла	зачтено/не зачтено
- научная и/или практическая работа	0 – 3 балла	зачтено/не зачтено
<b>Промежуточная аттестация:</b>		
- компьютерное тестирование	0 – 30 баллов	зачтено/не зачтено
<b>Итого за дисциплину</b>		
экзамен	0 - 100 баллов	Отлично, хорошо, удовлетворительно, неудовлетворительно

Полученный совокупный результат конвертируется в пятибалльную систему оценок в соответствии с таблицей:

100-балльная система	Пятибалльная система (оценка по дисциплине)
	экзамен
85 – 100 баллов	отлично
70 – 84 баллов	хорошо
55 – 69 баллов	удовлетворительно
0 – 54 баллов	неудовлетворительно

## 6. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Реализация программы предусматривает использование в процессе обучения следующих образовательных технологий:

- проблемная лекция;
- проектная деятельность;
- групповые дискуссии;
- анализ ситуаций и имитационных моделей;
- преподавание дисциплины на основе результатов научных исследований;
- поиск и обработка информации с использованием сети Интернет;
- дистанционные образовательные технологии;
- использование на лекционных занятиях видеоматериалов и наглядных пособий;
- самостоятельная работа в системе компьютерного тестирования.

## 7. ПРАКТИЧЕСКАЯ ПОДГОТОВКА

Практическая подготовка в рамках учебной дисциплины реализуется при проведении практических занятий и лабораторных работ, поскольку они предусматривают передачу учебной информации обучающимся, которая необходима для последующего выполнения практической работы.

## 8. ОРГАНИЗАЦИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ

При обучении лиц с ограниченными возможностями здоровья и инвалидов используются подходы, способствующие созданию безбарьерной образовательной среды: технологии дифференциации и индивидуального обучения, применение соответствующих методик по работе с инвалидами, использование средств дистанционного общения, проведение дополнительных индивидуальных консультаций по изучаемым теоретическим вопросам и практическим занятиям, оказание помощи при подготовке к промежуточной аттестации.

При необходимости рабочая программа дисциплины может быть адаптирована для обеспечения образовательного процесса лицам с ограниченными возможностями здоровья, в том числе для дистанционного обучения.

Учебные и контрольно-измерительные материалы представляются в формах, доступных для изучения студентами с особыми образовательными потребностями с учетом нозологических групп инвалидов:

Для подготовки к ответу на практическом занятии, студентам с ограниченными возможностями здоровья среднее время увеличивается по сравнению со средним временем подготовки обычного студента.

Для студентов с инвалидностью или с ограниченными возможностями здоровья форма проведения текущей и промежуточной аттестации устанавливается с учетом индивидуальных психофизических особенностей (устно, письменно на бумаге, письменно на компьютере, в форме тестирования и т.п.).

Промежуточная аттестация по дисциплине может проводиться в несколько этапов в форме рубежного контроля по завершению изучения отдельных тем дисциплины. При необходимости студенту предоставляется дополнительное время для подготовки ответа на зачете или экзамене.

Для осуществления процедур текущего контроля успеваемости и промежуточной аттестации обучающихся создаются, при необходимости, фонды оценочных средств, адаптированные для лиц с ограниченными возможностями здоровья и позволяющие оценить достижение ими запланированных в основной образовательной программе результатов обучения и уровень сформированности всех компетенций, заявленных в образовательной программе.

## 9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Характеристика материально-технического обеспечения дисциплины соответствует требованиям ФГОС ВО.

Материально-техническое обеспечение дисциплины при обучении с использованием традиционных технологий обучения.

<p>Наименование учебных аудиторий, лабораторий, мастерских, библиотек, спортзалов, помещений для хранения и профилактического обслуживания учебного оборудования и т.п.</p>	<p>Оснащенность учебных аудиторий, лабораторий, мастерских, библиотек, спортивных залов, помещений для хранения и профилактического обслуживания учебного оборудования и т.п.</p>
---	---

Наименование учебных аудиторий, лабораторий, мастерских, библиотек, спортзалов, помещений для хранения и профилактического обслуживания учебного оборудования и т.п.	Оснащенность учебных аудиторий, лабораторий, мастерских, библиотек, спортивных залов, помещений для хранения и профилактического обслуживания учебного оборудования и т.п.
<b>119071, г. Москва, Малый Калужский переулок, дом 1, строение 3, ауд.1440</b>	
аудитории для проведения занятий лекционного типа	комплект учебной мебели, технические средства обучения, служащие для представления учебной информации большой аудитории: – компьютерная техника (ноутбук/компьютер); – проектор; – экран.
аудитории для проведения практических занятий, выполнения лабораторных работ, занятий по практической подготовке, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	комплект учебной мебели, технические средства обучения, служащие для представления учебной информации большой аудитории: – компьютерная техника (ноутбук/компьютер); – проектор; – экран; – персональные компьютеры, подключенные к сети Интернет.
<b>Помещения для самостоятельной работы обучающихся</b>	<b>Оснащенность помещений для самостоятельной работы обучающихся</b>
читальный зал библиотеки:	– компьютерная техника, подключение к сети «Интернет»

Материально-техническое обеспечение учебной дисциплины при обучении с использованием электронного обучения и дистанционных образовательных технологий.

Необходимое оборудование	Параметры	Технические требования
Персональный компьютер/ноутбук/планшет, камера, микрофон, динамики, доступ в сеть Интернет	Веб-браузер	Версия программного обеспечения не ниже: Chrome 72, Opera 59, Firefox 66, Edge 79, Яндекс.Браузер 19.3
	Операционная система	Версия программного обеспечения не ниже: Windows 7, macOS 10.12 «Sierra», Linux
	Веб-камера	640x480, 15 кадров/с
	Микрофон	любой
	Динамики (колонки или наушники)	любые
	Сеть (интернет)	Постоянная скорость не менее 192 кБит/с

Технологическое обеспечение реализации программы осуществляется с использованием элементов электронной информационно-образовательной среды университета.

## 10. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

№ п/п	Автор(ы)	Наименование издания	Вид издания (учебник, УП, МП и др.)	Издательство	Год издания	Адрес сайта ЭБС или электронного ресурса (заполняется для изданий в электронном виде)	Количество экземпляров в библиотеке Университета
10.1 Основная литература, в том числе электронные издания							
1	Синаторов С.В.	Информационные технологии	Учебное пособие	М.: Флинта	2021	<a href="https://znanium.com/catalog/document?id=374932">https://znanium.com/catalog/document?id=374932</a>	-
2	Шитов В.Н.	Информатика и информационно-коммуникационные технологии в профессиональной деятельности	Учебное пособие	М: НИЦ ИНФРА-М	2022	<a href="https://znanium.com/catalog/document?id=388696">https://znanium.com/catalog/document?id=388696</a>	-
3	Немцова Т.И. и др.; под ред. Гагариной Л.Г.	Программирование на языке высокого уровня. Программирование на языке Object Pascal	Учебное пособие	М.: ИД ФОРУМ: НИЦ ИНФРА-М	2020	<a href="https://znanium.com/catalog/document?id=362746">https://znanium.com/catalog/document?id=362746</a>	-
4	Шуляк О.А.	Основы программирования	Учебно-методическая литература	М.: Флинта	2021	<a href="https://znanium.com/catalog/document?id=390158">https://znanium.com/catalog/document?id=390158</a>	-
5	Немцова Т.И., Голова С.Ю., Терентьев А.И.; под ред. Л.Г. Гагариной.	Программирование на языке высокого уровня. Программирование на языке С++	Учебное пособие	М.: ИД ФОРУМ: ИНФРА-М	2024	<a href="https://znanium.ru/catalog/document?id=432187">https://znanium.ru/catalog/document?id=432187</a>	-
10.2 Дополнительная литература, в том числе электронные издания							
1	Алексеев Е.Р., Чеснокова О.В.,	Free Pascal и Lazarus	Учебник	М: ДМК Пресс	2010	<a href="https://znanium.com/catalog/document?id=147218">https://znanium.com/catalog/document?id=147218</a>	-

	Кучер Т.В.						
2	Плотникова Н.Г.	Информатика и информационно-коммуникационные технологии (ИКТ)	Учебное пособие	М.: РИОР	2021	<a href="https://znanium.com/catalog/document?id=370445">https://znanium.com/catalog/document?id=370445</a>	-
3	Горбатов С.М., Тарасов Ю.С., Наумова М.Г.	Информационные технологии	Учебное пособие	М.: МИСиС	2016	<a href="https://znanium.com/catalog/document?id=371025">https://znanium.com/catalog/document?id=371025</a>	-
4	Федотова Е.Л.	Информационные технологии и системы	Учебное пособие	М.: Издательский Дом ФОРУМ	2022	<a href="https://znanium.com/catalog/document?id=386738">https://znanium.com/catalog/document?id=386738</a>	-
5	М. В. Огнева, Е. В. Кудрина	Программирование на языке С++: практический курс	Учебное пособие	М.: Издательство Юрайт	2022	<a href="https://urait.ru/bcode/492984">https://urait.ru/bcode/492984</a>	-
10.3 Методические материалы (указания, рекомендации по освоению дисциплины (модуля) авторов РГУ им. А. Н. Косыгина)							
1	Козлов А.М.	Технология объектно-ориентированного программирования на языке Pascal	Учебно-методическое пособие.	М.: ФГБОУ ВО «РГУ им. А.Н. Косыгина»	2020	локальная сеть университета	5
2	Козлов А.М.	Технология программирования на языке Pascal	Методические указания	М.: ФГБОУ ВО «РГУ им. А.Н. Косыгина»	2019	локальная сеть университета	5
3	Семенов А.А.	Основы объектно-ориентированного программирования в среде С++Builder	Методическое пособие	М.: ИИЦ МГУДТ	2010	локальная сеть университета	5

## 11. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

11.1. Ресурсы электронной библиотеки, информационно-справочные системы и профессиональные базы данных:

№ пп	Электронные учебные издания, электронные образовательные ресурсы
1.	ЭБС «Лань» <a href="http://www.e.lanbook.com/">http://www.e.lanbook.com/</a>
2.	«Znaniium.com» научно-издательского центра «Инфра-М» <a href="http://znaniium.com/">http://znaniium.com/</a>
3.	Электронные издания «РГУ им. А.Н. Косыгина» на платформе ЭБС «Znaniium.com» <a href="http://znaniium.com/">http://znaniium.com/</a>
4.	ЭБС «ИВИС» <a href="http://dlib.eastview.com/">http://dlib.eastview.com/</a>
5.	Образовательная платформа «ЮРАЙТ» <a href="https://urait.ru/">https://urait.ru/</a>
Профессиональные базы данных, информационные справочные системы	
1.	Scopus <a href="https://www.scopus.com">https://www.scopus.com</a> (международная универсальная реферативная база данных, индексирующая более 21 тыс. наименований научно-технических, гуманитарных и медицинских журналов, материалов конференций примерно 5000 международных издательств);
2.	Научная электронная библиотека eLIBRARY.RU <a href="https://elibrary.ru">https://elibrary.ru</a> (крупнейший российский информационный портал в области науки, технологии, медицины и образования);
3.	База данных в мире Academic Search Complete - обширная полнотекстовая научно-исследовательская. Содержит полные тексты тысяч рецензируемых научных журналов по химии, машиностроению, физике, биологии. <a href="http://search.ebscohost.com">http://search.ebscohost.com</a>

11.2. Перечень программного обеспечения

№п/п	Программное обеспечение	Реквизиты подтверждающего документа/ Свободно распространяемое
1.	Windows 10 Pro, MS Office 2019	контракт № 18-ЭА-44-19 от 20.05.2019
2.	Lazarus — открытая среда разработки программного обеспечения на языке Object Pascal для компилятора Free Pascal.	Свободно распространяемое на условиях GNU General Public License.
3.	Microsoft Visual Studio	контракт № 18-ЭА-44-19 от 20.05.2019
4.	Embarcadero C++Builder RAD Studio Professional Academic Concurrent License	№ 15-02.01-2459 от 21.12.2021 Embarcadero License Certificate: #546431, #546432, #546433, #546434, #546435
5.	Code::Blocks — свободная кроссплатформенная среда разработки на C++.	Свободно распространяемое на условиях GNU General Public License v.3.
6.	Visual Studio Community	Свободно распространяемая среда разработки.
7.	Visual Studio Code	Свободно распространяемая среда разработки.

**ЛИСТ УЧЕТА ОБНОВЛЕНИЙ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ**

В рабочую программу учебной дисциплины внесены изменения/обновления и утверждены на заседании кафедры:

<b>№ пп</b>	<b>год обновления РПД</b>	<b>характер изменений/обновлений с указанием раздела</b>	<b>номер протокола и дата заседания кафедры</b>