

МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение
высшего образования

«Кузбасский государственный технический университет имени Т. Ф. Горбачева»

Институт профессионального образования



ПОДПИСАНО ЭП КУЗГТУ

Подразделение: ректорат

Должность: проректор по среднему
профессиональному образованию

Дата: 15.04.2022 08:58:03

Попов Иван Павлович

Рабочая программа дисциплины

Основы алгоритмизации и программирования

Специальность «10.02.05 Обеспечение информационной безопасности автоматизированных систем»

Присваиваемая квалификация
"Техник по защите информации"

Формы обучения
очная

Кемерово 2022 г.

Рабочую программу составил:

ПОДПИСАНО ЭП КУЗГТУ

Подразделение: учебно-методическое управление

Должность: начальник управления

Дата: 14.04.2022 03:25:34

Прокопенко Евгения Викторовна

Рабочая программа обсуждена на заседании кафедры информационной безопасности

Протокол № 3/1 от 15.04.2022

ПОДПИСАНО ЭП КУЗГТУ

Подразделение: учебно-методическое управление

Должность: начальник управления

Дата: 15.04.2022 01:26:45

Прокопенко Евгения Викторовна

Согласовано цикловой-методической комиссией по направлению подготовки (специальности)
10.02.05 Обеспечение информационной безопасности автоматизированных систем

Протокол № 4/1 от 15.04.2022

ПОДПИСАНО ЭП КУЗГТУ

Подразделение: учебно-методическое управление

Должность: начальник управления

Дата: 15.04.2022 03:33:16

Прокопенко Евгения Викторовна

Согласовано заместителем директора по УР ИПО

ПОДПИСАНО ЭП КУЗГТУ

Подразделение: учебно-методическое управление

Должность: Заместитель директора по учебной работе

Дата: 15.04.2022 03:33:16

Полуэктова Наталья Сергеевна

Согласовано заместителем директора по МР ИПО

ПОДПИСАНО ЭП КУЗГТУ

Подразделение: учебно-методическое управление

Должность: Заместитель директора по методической работе

Дата: 15.04.2022 03:33:16

Сьянова Татьяна Юрьевна

1. Общая характеристика рабочей программы дисциплины

1.1 Место дисциплины в структуре основной образовательной программы

Учебная дисциплина «Основы алгоритмизации и программирования» является обязательной частью общепрофессионального цикла основной образовательной программы в соответствии с ФГОС по специальности 10.02.05 «Обеспечение информационной безопасности автоматизированных систем». Учебная дисциплина «Основы алгоритмизации и программирования» обеспечивает формирование профессиональных и общих компетенций по всем видам деятельности ФГОС по специальности 10.02.05 «Обеспечение информационной безопасности автоматизированных систем».

1.2 Цель и планируемые результаты освоения дисциплины, соотнесенные с планируемыми результатами освоения образовательной программы

Освоение дисциплины направлено на формирование:

общих компетенций:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

Знать: способы решения задач профессиональной деятельности;

Уметь: определять способы решения задач профессиональной деятельности;

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

Знать: типы данных;

Уметь: проводить поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

Знать: способы планирования и реализации собственного профессионального и личностного развития;

Уметь: планировать и реализовывать собственное профессиональное и личностное развитие;

профессиональных компетенций:

ПК 2.1. Осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации.

Знать: принципы установки и настройки отдельных программных, программно-аппаратных средств защиты информации; Уметь: осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации; Иметь практический опыт:

установки и настройки отдельных программных, программно-аппаратных средств защиты информации;

ПК 2.2. Обеспечивать защиту информации в автоматизированных системах отдельными программными, программно-аппаратными средствами.

Знать: базовые конструкции изучаемых языков программирования; способы защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами; Уметь: обеспечивать защиту информации в автоматизированных системах

отдельными программными, программно-аппаратными средствами; Иметь практический опыт:

обеспечения защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами;

ПК 2.3. Осуществлять тестирование функций отдельных программных и программно-аппаратных средств защиты информации.

Знать: базовые конструкции изучаемых языков программирования; интегрированные среды программирования на изучаемых языках; Уметь: работать в среде программирования;

использовать языки программирования высокого уровня; Иметь практический опыт: проведения тестирования функций отдельных программных и программно-аппаратных средств защиты информации;

ПК 2.4. Осуществлять обработку, хранение и передачу информации ограниченного доступа.

Знать: типы данных; принципы обработки, хранения и передачи информации ограниченного доступа; Уметь: работать в среде программирования; использовать языки программирования

высокого уровня; Иметь практический опыт: обработки, хранения и передачи информации;

ПК 2.6. Осуществлять регистрацию основных событий в автоматизированных (информационных) системах, в том числе с использованием программных и программно-аппаратных средств обнаружения, предупреждения и ликвидации последствий компьютерных атак.

Знать: интегрированные среды программирования на изучаемых языках; Уметь: использовать языки программирования высокого уровня; Иметь практический опыт: регистрации основных событий в автоматизированных (информационных) системах, в том числе с использованием программных и программно-аппаратных средств обнаружения, предупреждения и ликвидации последствий компьютерных атак;

В результате освоения дисциплины обучающийся в общем по дисциплине должен

Знать:

- способы решения задач профессиональной деятельности;
- типы данных;
- способы планирования и реализации собственного профессионального и личностного развития;
- принципы установки и настройки отдельных программных, программно-аппаратных средств защиты информации;

- базовые конструкции изучаемых языков программирования; способы защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами;

- базовые конструкции изучаемых языков программирования; интегрированные среды программирования на изучаемых языках;

- типы данных; принципы обработки, хранения и передачи информации ограниченного доступа;

- интегрированные среды программирования на изучаемых языках;

Уметь:

- определять способы решения задач профессиональной деятельности;

- проводить поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;

- планировать и реализовывать собственное профессиональное и личностное развитие;

- осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации;

- обеспечивать защиту информации в автоматизированных системах отдельными программными, программно-аппаратными средствами;

- работать в среде программирования; использовать языки программирования высокого уровня;

- использовать языки программирования высокого уровня;

Иметь практический опыт:

- установки и настройки отдельных программных, программно-аппаратных средств защиты информации;

- обеспечения защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами;

- проведения тестирования функций отдельных программных и программно-аппаратных средств защиты информации;

- обработки, хранения и передачи информации;

- регистрации основных событий в автоматизированных (информационных) системах, в том числе с использованием программных и программно-аппаратных средств обнаружения, предупреждения и ликвидации последствий компьютерных атак;

2. Структура и содержание дисциплины

2.1 Объем дисциплины и виды учебной работы

Форма обучения	Количество часов		
	ОФ	ЗФ	ОЗФ
Курс 2 / Семестр 3			
Объем дисциплины	84		
в том числе:			
<i>лекции, уроки</i>	32		
<i>лабораторные работы</i>			
<i>практические занятия</i>	32		

Форма обучения	Количество часов		
	ОФ	ЗФ	ОЗФ
Консультации			
Самостоятельная работа	20		
Промежуточная аттестация			
Индивидуальное проектирование			
Форма промежуточной аттестации			
Курс 2 / Семестр 4			
Объем дисциплины	126		
в том числе:			
<i>лекции, уроки</i>	32		
<i>лабораторные работы</i>			
<i>практические занятия</i>	60		
Консультации	6		
Самостоятельная работа	22		
Промежуточная аттестация	6		
Индивидуальное проектирование			
Форма промежуточной аттестации	экзамен		

2.2 Тематический план и содержание дисциплины

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах
3 семестр		
Раздел 1. Основные принципы алгоритмизации и программирования		
Тема 1.1. Основные понятия алгоритмизации		
<i>Лекции.</i>		
	Лекция 1.1.1. Понятие алгоритма и его свойства. Типы алгоритмов. Способы описания алгоритмов. Базовые алгоритмические структуры: линейные, разветвляющиеся, циклические.	1
	Лекция 1.1.2. Основные базовые типы данных и их характеристика. Основы алгебры логики. Логические операции и логические функции.	1
Тема 1.2. Принципы разработки алгоритмов		
<i>Лекции</i>		
	Лекция 1.2.1. Принципы построения алгоритмов: использование базовых структур, метод последовательной детализации, сборочный метод. Разработка алгоритмов сложной структуры.	2
<i>Лабораторные занятия</i>		

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах
	Практическое занятие 1.2.1. Разработка линейных алгоритмов и алгоритмов ветвления.	1
	Практическое занятие 1.2.2. Разработка циклических алгоритмов.	1
	Практическое занятие 1.2.3. Разработка алгоритмов шифрования.	1
<i>Самостоятельная работа обучающихся</i>		
	1.2.1. Разработка алгоритмов различного типа	1
Тема 1.3. Языки и системы программирования		
<i>Лекции</i>		
	Лекция 1.3.1. Классификация языков программирования. Понятие интегрированной среды программирования. Способы классификации систем программирования. Перечень и назначение модулей системы программирования.	2
Тема 1.4. Парадигмы программирования		
<i>Лекции</i>		
	Лекция 1.4.1. Этапы разработки программ: системный анализ, алгоритмизация, программирование, отладка, сопровождение. Характеристика и задачи каждого этапа. Принципы структурного программирования: использование базовых структур, декомпозиция базовых структур. Понятия основных элементов ООП: объекты, классы, методы. Свойства ООП: наследование, инкапсуляция, полиморфизм. Принципы модульного программирования.	4
<i>Лабораторные занятия</i>		
	Практическое занятие 1.4.1. Подготовка конспекта по теме «Типы приложений»	1
Тема 1.5. Принципы отладки и тестового контроля		
<i>Лекции</i>		
	Лекция 1.5.1. Понятие отладки. Понятие тестового контроля и набора тестов. Проверка граничных условий, ветвей алгоритма, ошибочных исходных данных. Функциональное и структурное тестирование.	2
	Контрольная работа : Этапы разработки программ.	2
Раздел 2. Язык программирования		
Тема 2.1. Характеристика языка		
<i>Лекции</i>		

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах
	Лекция 2.1.1. История и особенности языка. Области применения. Характеристика системы программирования. Процесс трансляции и выполнения программы.	2
Тема 2.2. Элементы языка. Простые типы данных		
<i>Лекции</i>		
	Лекция 2.2.1. Алфавит и лексика языка. Структура программы. Типы данных языка программирования. Переменные и их описания. Операции с переменными и константами. Правила записи выражений и операций. Организация ввода/вывода данных.	2
<i>Лабораторные занятия</i>		
	Практическое занятие 2.2.1. Знакомство с инструментальной средой программирования	1
<i>Самостоятельная работа обучающихся</i>		
	2.2.1. Использование программного обеспечения для разработки алгоритмов: освоение возможностей компилятора	1
	2.2.2. Составление программ по теме «Линейные программы»	1
Тема 2.3. Базовые конструкции структурного программирования		
<i>Лекции</i>		
	Лекция 2.3.1. Организация ветвлений. Операторы циклов (с предусловием, с постусловием, с параметром). Операторы передачи управления.	2
<i>Лабораторные занятия</i>		
	Практическое занятие 2.3.1. Разработка программ разветвляющейся структуры.	2
	Практическое занятие 2.3.2. Разработка программ с использованием цикла с предусловием.	2
	Практическое занятие 2.3.3. Разработка программ с использованием цикла с постусловием.	2
	Практическое занятие 2.3.4. Разработка программ с использованием цикла с параметром.	2
<i>Самостоятельная работа обучающихся</i>		
	2.3.1. Составление программ по теме «Разветвляющиеся структуры»	1
	2.3.2. Составление программ по теме «Циклы с предусловием»	1
	2.3.3. Составление программ по теме «Циклы с постусловием»	1
	2.3.4. Составление программ по теме «Циклы с параметром»	1

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах
Тема 2.4. Работа с массивами и указателями. Структурные типы данных		
<i>Лекции</i>		
	Лекция 2.4.1. Одномерные и многомерные массивы, их формирование, сортировка, обработка. Указатели и операции над ними.	4
	Лекция 2.4.2. Работа со строками. Структуры и объединения.	2
<i>Лабораторные занятия</i>		
	Практическое занятие 2.4.1. Разработка программ с использованием одномерных массивов и указателей.	2
	Практическое занятие 2.4.2. Сортировка одномерных массивов.	2
	Практическое занятие 2.4.3. Разработка программ с использованием двумерных массивов.	1
	Практическое занятие 2.4.4. Сортировка двумерных массивов.	1
	Практическое занятие 2.4.5. Разработка программ с использованием структур.	1
	Практическое занятие 2.4.6. Разработка программ с использованием строк.	1
	Контрольная работа : Решение задач на базовые конструкции.	1
<i>Самостоятельная работа обучающихся</i>		
	2.4.1. Составление программ по теме «Одномерные массивы»	0,5
	2.4.2. Составление программ по теме «Многомерные массивы»	0,5
	2.4.3. Составление программ по теме «Указатели»	1
	2.4.4. Составление программ по теме «Сортировка массивов различными методами»	1
	2.4.5. Составление программ по теме «Работа со строками»	1
	2.4.6. Составление программ по теме «Работа со структурами».	1
Тема 2.5. Процедуры и функции		
<i>Лекции</i>		
	Лекция 2.5.1. Определение процедур и функций. Области видимости. Глобальные и локальные переменные. Обращение к процедурам и функциям.	1
	Лекция 2.5.2. Использование библиотечных функций. Рекурсивное определение функций. Шаблоны функций.	1
<i>Лабораторные занятия</i>		

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах
	Практическое занятие 2.5.1. Разработка программ с использованием функций.	2
	Практическое занятие 2.5.2. Разработка программ с использованием рекурсивных функций.	2
<i>Самостоятельная работа обучающихся</i>		
	2.5.1. Составление программ по теме «Нерекурсивные функции»	1
	2.5.2. Составление программ по теме «Рекурсивные функции»	1
Тема 2.6. Работа с файлами		
<i>Лекции</i>		
	Лекция 2.6.1. Файловый ввод/вывод. Организация обмена данными между программой и внешними устройствами компьютера. Ввод и вывод текстовой информации. Неформатированный ввод/вывод данных. Дополнительные операции с файлами.	4
<i>Лабораторные занятия</i>		
	Практическое занятие 2.6.1. Разработка программ работы со структурированными файлами.	2
	Практическое занятие 2.6.2. Разработка программ работы с текстовыми файлами.	2
	Практическое занятие 2.6.3. Разработка программ работы с неструктурированными файлами.	2
<i>Самостоятельная работа обучающихся</i>		
	2.6.1. Составление программ по теме «Работа с файлами»	2
	2.6.2. Составление программ по теме «Работа с тестовыми файлами»	2
	2.6.3. Составление программ по теме «Работа с типизированными файлами»	2
Итого		84
4 семестр		
Раздел 3. Основы объектно-ориентированного программирования		
Тема 3.1. Класс - как механизм создания объектов		
<i>Лекции</i>		
	Лекция 3.1.1. Понятия: класс, объект, свойства объекта, методы. Синтаксис объявления класса. Описание объектов.	3
	Лекция 3.1.2. Спецификаторы доступа (private, public, protected). Описание функций-членов класса. Принцип инкапсуляции.	3

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах
<i>Лабораторные занятия</i>		
	Практическое занятие 3.1.1. Организация классов и принцип инкапсуляции.	6
	Практическое занятие 3.1.2. Разработка приложений с использованием классов.	6
<i>Самостоятельная работа обучающихся</i>		
	3.1.1. Разработка приложений с использованием классов.	6
Тема 3.2. Принципы наследования и полиморфизма		
<i>Лекции</i>		
	Лекция 3.2.1. Механизм наследования для формирования иерархии классов. Формат объявления класса потомка. Режим доступа.	3
	Лекция 3.2.2. Примеры организации классов-наследников	3
<i>Лабораторные занятия</i>		
	Практическое занятие 3.2.1. Программная реализация принципов наследования.	6
	Практическое занятие 3.2.2. Программная реализация принципов полиморфизма	6
<i>Самостоятельная работа обучающихся</i>		
	3.2.1. Разработка классов потомков	4
	3.2.2. Реализация механизма перегрузки	4
Тема 3.3. Понятия деструктора и конструктора		
<i>Лекции</i>		
	Лекция 3.3.1. Назначение и свойства конструкторов, деструкторов. Их описание. Вызов в программе конструкторов, деструкторов. Примеры программ с конструкторами и деструкторами.	4
<i>Лабораторные занятия</i>		
	Практическое занятие 3.3.1. Разработка конструкторов и деструкторов.	6
<i>Самостоятельная работа обучающихся</i>		
	3.3.1. Составление программ по теме «Конструкторы и деструкторы».	4
Раздел 4. Модульное программирование		
Тема 4.1. Понятие модульного программирования		

Наименование разделов и тем	Содержание учебного материала и формы организации деятельности обучающихся	Объем в часах
<i>Лекции</i>		
	Лекция 4.1.1. Модульное программирование как метод разработки программ. Программный модуль и его основные характеристики. Типовая структура программного модуля. Инкапсуляция в модулях.	4
	Лекция 4.1.2. Порядок разработки программного модуля. Связность модулей. Ошибки периода исполнения и логические ошибки в программах. Обработка ошибок. Исключительные ситуации. Организация обработки исключительных ситуаций.	4
Тема 4.2. Разработка приложений		
<i>Лекции</i>		
	Лекция 4.2.1. Среда разработки приложений. Архитектура оконных приложений. Конфигурации для создания консольных и оконных приложений.	4
	Лекция 4.2.2. Разработка приложений как многомодульного проекта.	4
<i>Лабораторные занятия</i>		
	Практическое занятие 4.2.1. Разработка многомодульных приложений.	
<i>Самостоятельная работа обучающихся</i>		
	4.2.1. Разработка многомодульных приложений.	4
Консультации		6
Промежуточная аттестация в форме экзамена		6
Итого		126

3 Материально-техническое и учебно-методическое обеспечение дисциплины (модуля)

3.1 Специальные помещения для реализации программы

Для реализации программы учебной дисциплины должны быть предусмотрены следующие специальные помещения:

1. Специальное помещение № 1406 представляет собой учебную аудиторию для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.

Перечень основного оборудования:

Комплект мебели (столы и стулья). Проектор. Персональный компьютер.

Перечень программного обеспечения: Libre Office. Mozilla Firefox. Google Chrome. 7-zip .Microsoft Windows. ESET NOD32 Smart Security Business Edition. Kaspersky Endpoint Security. Браузер Спутник.

2. Специальное помещение № 1147 представляет собой помещение для групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы, мастерские и лаборатории, оснащенные оборудованием, техническими средствами обучения и материалами, учитывающими требования международных, национальных и межгосударственных стандартов в области защиты информации.

Перечень основного оборудования:

Специализированная мебель (столы и стулья); Коммутаторы, Металлические рольставни с пружинным механизмом, белые 1650мм*2270мм; Сейф металлический; Системные блоки ITS (i3-10100/H410M/8 Gb/SSD 240Gb/БП AA500W); Точка доступа D-link; Мониторы 23.6" AOC 24B1H VA 1920x1080 (16:9), 250кд/м2, 5мс, VGA, HDMI, черные; Системные блоки MasteroMiddleMC05, IntelCorei510400 2.9GHz, 8GbRAM, 240GbSSD, DOS, программно-аппаратный комплекс для обнаружения компьютерных атак VipNet, средство доверенной загрузки (СДЗ) Соболев

3. Помещение для самостоятельной работы обучающихся:

Специальное помещение № 1237 представляет собой помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к информационно-телекоммуникационной сети Интернет и обеспечением доступа в электронную информационно-образовательную среду образовательной организации:

Перечень основного оборудования:

Комплект мебели (столы и стулья). Персональные компьютеры. Коммутатор AlliedTelesynLayer 2 SmartSwitch,

Перечень программного обеспечения: LibreOffice. MozillaFirefox. Google Chrome. 7-zip .Microsoft Windows. ESET NOD32 Smart Security Business Edition. Kaspersky Endpoint Security. БраузерСпутник.

4. Помещение для самостоятельной работы обучающихся:

Специальное помещение № 1211 представляет собой помещения для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к информационно-телекоммуникационной сети Интернет и обеспечением доступа в электронную информационно-образовательную среду образовательной организации:

Перечень основного оборудования:

Специализированная мебель (столы и стулья); компьютерная техника с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду КузГТУ, в том числе:

проектор, экран настенный моторизованный.

Перечень программного обеспечения: LibreOffice. Mozilla Firefox. Google Chrome. 7-zip .Microsoft Windows. ESET NOD32 Smart Security Business Edition. Kaspersky Endpoint Security. БраузерСпутник.

5. Специальное помещение №1146 представляет собой лабораторию информационных технологий, сетей и систем передачи информации, программирования и баз данных, оснащенную рабочими местами на базе вычислительной техники, подключенными к локальной вычислительной сети и информационно-телекоммуникационной сети "Интернет"; программным обеспечением сетевого оборудования; обучающим программным обеспечением; эмуляторами активного сетевого оборудования; программным обеспечением межсетевого экранирования и мониторинга технического состояния активного сетевого оборудования.

Перечень основного оборудования:

Комплект мебели (столы и стулья).

Мультимедиа-проектор BenQ MP721C; Ноутбук AcerAspire5102WLM.; Проектор Aser P1383W с кронштейном, видео кабелем 20 м; Сейф металлический; Сплинг-система RODA RS\RU-A 18B серия Arctic; Сплинг-система RU-A07B серия Arctic; Экран настенный рулонный Projecta ProScreen 183*240 см.; Системный блок МК Office (Intel Core i3/4Гб/500Гб); IP-камера ZQ-IPC3-DAS-36VI Камера внутр., купольная, 1/2.8 "SONY; Моноблок Powercool, Россия; Многофункциональное устройство (МФУ) PANTUM M6500; Принтер лазерный Kyosera Ecosys P2040dn.A4 ч\б) 1200*1200dpi. дуплэкс, сетевой; Перечень программного обеспечения: Libre Office. Mozilla Firefox. Google Chrome. 7-zip .Microsoft Windows. ESET NOD32 Smart Security Business Edition. Kaspersky Endpoint Security. Браузер Спутник.

3.2 Информационное обеспечение реализации программы

3.2.1 Основная литература

1. Семакин, И. Г. Основы алгоритмизации и программирования : учебник для образовательных организаций, реализующих программы среднего профессионального образования по специальностям "Информационные системы и программирование", "Сетевое и системное администрирование", "Обеспечение информационной безопасности автоматизированных систем", "Обеспечение информационной / И. Г. Семакин, А. П. Шестаков ; И. Г. Семакин, А. П. Шестаков. - 4-е изд., стер. - Москва : Академия, 2020. - с. - (Профессиональное образование). - URL: <https://academia-moscow.ru/reader/?id=471483> (дата обращения: 26.09.2023). - Текст : электронный.

3.2.2 Дополнительная литература

1. Нагаева, И. А. Основы алгоритмизации и программирования / И. А. Нагаева, И. А. Кузнецов. – Москва, Берлин : Директ-Медиа, 2021. – 168 с. – ISBN 9785449916129. – URL: http://biblioclub.ru/index.php?page=book_red&id=598404 (дата обращения: 26.09.2023). – Текст : электронный.
2. Гуриков, С. Р. Введение в программирование на языке Visual Basic for Applications (VBA) : Учебное пособие / С. Р. Гуриков ; Московский технический университет связи и информатики. – Москва : НИЦ ИНФРА-М, 2021. – 317 с. – ISBN 978-5-16-015995-9. – URL: <http://znanium.com/catalog/document?id=365678> (дата обращения: 26.09.2023). – Текст : электронный.
3. Казанский, А. А. Программирование на visual c#: учебное пособие для СПО / Казанский А. А.. – 2-е изд., пер. и доп. – Москва : Юрайт, 2020. – 192 с. – ISBN 978-5-534-14130-6. – URL: <https://urait.ru/book/programmirovaniye-na-visual-c-2013-467844> (дата обращения: 26.09.2023). – Текст : электронный.
4. Маркин, А. В. Программирование на sql.: учебное пособие для СПО / Маркин А. В.. – Москва : Юрайт, 2020. – 435 с. – ISBN 978-5-534-11093-7. – URL: <https://urait.ru/book/programmirovaniye-na-sql-456926> (дата обращения: 26.09.2023). – Текст : электронный.
5. Огнева, М. В. Программирование на языке c++: практический курс.: учебное пособие для СПО / Огнева М. В., Кудрина Е. В.. – Москва : Юрайт, 2020. – 335 с. – ISBN 978-5-534-05780-5. – URL: <https://urait.ru/book/programmirovaniye-na-yazyke-s-prakticheskiy-kurs-454231> (дата обращения: 26.09.2023). – Текст : электронный.
6. Голицына, О. Л. Основы алгоритмизации и программирования : Учебное пособие / О. Л. Голицына, И. И. Попов. – Москва : НИЦ ИНФРА-М, 2021. – 431 с. – ISBN 978-5-00091-570-7. – URL: <https://znanium.com/catalog/document?id=361010> (дата обращения: 26.09.2023). – Текст : электронный.
7. Колдаев, В. Д. Основы алгоритмизации и программирования : Учебное пособие / В. Д. Колдаев, Л. Г. Гагарина. – Москва : НИЦ ИНФРА-М, 2022. – 414 с. – ISBN 978-5-8199-0733-7. – URL: <https://znanium.com/catalog/document?id=378685> (дата обращения: 26.09.2023). – Текст : электронный.
8. Трофимов, В. В. Основы алгоритмизации и программирования.: учебник для СПО / Трофимов В. В., Павловская Т. А. ; Под ред. Трофимова В.В.. – Москва : Юрайт, 2021. – 137 с. – ISBN 978-5-534-07321-8. – URL: <https://urait.ru/book/osnovy-algoritmizacii-i-programmirovaniya-473347> (дата обращения: 26.09.2023). – Текст : электронный.

3.2.3 Методическая литература

1. Основы алгоритмизации и программирования : методические материалы для обучающихся направления подготовки 10.02.05 "Обеспечение информационной безопасности автоматизированных систем" / Кузбасский государственный технический университет им. Т. Ф. Горбачева ; Кафедра информационной безопасности, составители: Е. В. Прокопенко, А. В. Медведев, А. Г. Киренберг. – Кемерово : КузГТУ, 2020. – 15 с. – URL: <http://library.kuzstu.ru/meto.php?n=9962> (дата обращения: 26.09.2023). – Текст : электронный.

3.2.4 Интернет ресурсы

1. ЭИОС КузГТУ:
 - а) Электронная библиотека КузГТУ. – Текст: электронный // Научно-техническая библиотека Кузбасского государственного технического университета им. Т. Ф. Горбачева : сайт. – Кемерово, 2001 - . - URL: <https://elib.kuzstu.ru/> . – Текст: электронный.
 - б) Портал.КузГТУ : Автоматизированная Информационная Система (АИС) : [сайт] / Кузбасский государственный технический университет им. Т. Ф. Горбачева. – Кемерово : КузГТУ, [б. г.]. – URL: <https://portal.kuzstu.ru/>. – Режим доступа: для авториз. пользователей. – Текст: электронный.
 - в) Электронное обучение : [сайт] / Кузбасский государственный технический университет им. Т. Ф. Горбачева. – Кемерово : КузГТУ, [б. г.]. – URL: <https://el.kuzstu.ru/> . – Режим доступа: для авториз. пользователей КузГТУ. – Текст: электронный.
2. ФСТЭК России : Федеральная служба по техническому и экспортному контролю : официальный сайт / ФАУ «ГНИИИ ПТЗИ ФСТЭК России». – Москва, 2004 - . - URL: www.fstec.ru. – Текст: электронный.
3. SecurityLab.ru : информационный портал по безопасности : сайт. – Москва. – URL: <https://www.securitylab.ru/> . – Текст: электронный.

4. Департамент образования Вологодской области : официальный сайт. - Вологда. - URL: <http://depobr.gov35.ru/> . - Текст: электронный.
5. BIOMETRICS.RU : Российский биометрический портал : сайт. - Москва, 2000 - . - URL: www.biometrics.ru . - Текст: электронный.
6. InformationSecurity/Информационная безопасность : сайт. - Москва. - URL: <http://www.itsec.ru>. - Текст: электронный.
7. eLIBRARY.RU : научная электронная библиотека : сайт. - Москва, 2000 - . - URL: <https://elibrary.ru>. - Режим доступа: для зарегистрир. пользователей. - Текст: электронный.
8. Гарант. ру : информационно-правовой портал : сайт. - Москва, 1990 - . - URL: <https://www.garant.ru/> . - Текст: электронный.
9. КонсультантПлюс : компьютерная справочно-правовая система : сайт. - Москва, 1992 - . - URL: www.consultant.ru . - Текст: электронный.
10. Единое окно доступа к образовательным ресурсам : информационная система : сайт / ФГАУ ГНИИ ИТТ «Информика» . - Москва, 2005 - . - URL: <http://window.edu.ru/> . - Текст: электронный.
11. Российское образование. Федеральный образовательный портал : сайт / ФГАОУ ДПО ЦРГОП и ИТ. - Москва, 2002 - . - URL: www.edu.ru . - Текст: электронный.

4. Организация самостоятельной работы обучающихся

Самостоятельная работа обучающихся осуществляется в объеме, установленном в разделе 2 настоящей программы дисциплины (модуля).

Для самостоятельной работы обучающихся предусмотрены специальные помещения, оснащенные компьютерной техникой с возможностью подключения к информационно-телекоммуникационной сети "Интернет" с обеспечением доступа в электронную информационно-образовательную среду КузГТУ.

5. Фонд оценочных средств для проведения текущего контроля, промежуточной аттестации обучающихся по дисциплине

5.1 Паспорт фонда оценочных средств

Планируемые результаты обучения по дисциплине.

Дисциплина направлена на формирование следующих компетенций выпускника:

5.1 Паспорт фонда оценочных средств

№	Наименование разделов дисциплины	Содержание (темы) раздела	Код компетенции	Знания, умения, практический опыт, необходимые для формирования соответствующей компетенции	Форма текущего контроля знаний, умений, практического опыта, необходимых для формирования соответствующей компетенции

1	Раздел 1. Основные принципы алгоритмизации и программирования	Тема 1.1. Основные понятия алгоритмизации Тема 1.2. Принципы разработки алгоритмов Тема 1.3. Языки и системы программирования Тема 1.4. Парадигмы программирования Тема 1.5. Принципы отладки и тестового контроля	ОК 01.	Знать: способы решения задач профессиональной деятельности; Уметь: определять способы решения задач профессиональной деятельности;	опрос обучающихся по контрольным вопросам, защита отчетов по лабораторным заданиям, тестирование, контрольная работа
			ОК 02.	Знать: типы данных; Уметь: проводить поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;	
			ОК 03.	Знать: способы планирования и реализации собственного профессионального и личностного развития; Уметь: планировать и реализовывать собственное профессиональное и личностное развитие;	
			ПК 2.1.	Знать: принципы установки и настройки отдельных программных, программно-аппаратных средств защиты информации; Уметь: осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации; Иметь опыт: установки и настройки отдельных программных, программно-аппаратных средств защиты информации;	
			ПК 2.2.	Знать: базовые конструкции изучаемых языков программирования; способы защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами; Уметь: обеспечивать защиту информации в автоматизированных системах отдельными программными, программно-аппаратными средствами; Иметь опыт: обеспечения защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами;	
			ПК 2.3.	Знать: базовые конструкции изучаемых языков программирования; интегрированные среды программирования на изучаемых языках; Уметь: работать в среде программирования; использовать языки программирования высокого уровня; Иметь опыт: проведения тестирования функций отдельных программных и программно-аппаратных средств защиты информации;	
			ПК 2.4.	Знать: типы данных; принципы обработки, хранения и передачи информации ограниченного доступа; Уметь: работать в среде программирования; использовать языки программирования высокого уровня; Иметь опыт: обработки, хранения и передачи информации;	
			ПК 2.6.	Знать: интегрированные среды программирования на изучаемых языках; Уметь: использовать языки программирования высокого уровня; Иметь опыт: регистрации основных событий в автоматизированных (информационных) системах, в том числе с использованием программных и программно-аппаратных средств обнаружения, предупреждения и ликвидации последствий компьютерных атак;	

2	Раздел 2. Язык программирования	Тема 2.1. Характеристика языка Тема 2.2. Элементы языка. Простые типы данных Тема 2.3. Базовые конструкции структурного программирования Тема 2.4. Работа с массивами и указателями. Структурные типы данных Тема 2.5. Процедуры и функции Тема 2.6. Работа с файлами	ОК 01.	Знать: способы решения задач профессиональной деятельности; Уметь: определять способы решения задач профессиональной деятельности;	опрос обучающихся по контрольным вопросам, защита отчетов по лабораторным заданиям, тестирование, контрольная работа
			ОК 02.	Знать: типы данных; Уметь: проводить поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;	
			ПК 2.1.	Знать: принципы установки и настройки отдельных программных, программно-аппаратных средств защиты информации; Уметь: осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации; Иметь опыт: установки и настройки отдельных программных, программно-аппаратных средств защиты информации;	
			ПК 2.2.	Знать: базовые конструкции изучаемых языков программирования; способы защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами; Уметь: обеспечивать защиту информации в автоматизированных системах отдельными программными, программно-аппаратными средствами; Иметь опыт: обеспечения защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами;	
			ПК 2.3.	Знать: базовые конструкции изучаемых языков программирования; интегрированные среды программирования на изучаемых языках; Уметь: работать в среде программирования; использовать языки программирования высокого уровня; Иметь опыт: проведения тестирования функций отдельных программных и программно-аппаратных средств защиты информации;	
			ПК 2.4.	Знать: типы данных; принципы обработки, хранения и передачи информации ограниченного доступа; Уметь: работать в среде программирования; использовать языки программирования высокого уровня; Иметь опыт: обработки, хранения и передачи информации;	
			ПК 2.6.	Знать: интегрированные среды программирования на изучаемых языках; Уметь: использовать языки программирования высокого уровня; Иметь опыт: регистрации основных событий в автоматизированных (информационных) системах, в том числе с использованием программных и программно-аппаратных средств обнаружения, предупреждения и ликвидации последствий компьютерных атак;	

3	Раздел 3. Основы объектно-ориентированного программирования	Тема 3.1. Класс - как механизм создания объектов Тема 3.2. Принципы наследования и полиморфизма Тема 3.3. Понятия деструктора и конструктора	ОК 01.	Знать: способы решения задач профессиональной деятельности; Уметь: определять способы решения задач профессиональной деятельности;	опрос обучающихся по контрольным вопросам, защита отчетов по лабораторным заданиям, тестирование
			ОК 02.	Знать: типы данных; Уметь: проводить поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;	
			ПК 2.1.	Знать: принципы установки и настройки отдельных программных, программно-аппаратных средств защиты информации; Уметь: осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации; Иметь опыт: установки и настройки отдельных программных, программно-аппаратных средств защиты информации;	
			ПК 2.2.	Знать: базовые конструкции изучаемых языков программирования; способы защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами; Уметь: обеспечивать защиту информации в автоматизированных системах отдельными программными, программно-аппаратными средствами; Иметь опыт: обеспечения защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами;	
			ПК 2.3.	Знать: базовые конструкции изучаемых языков программирования; интегрированные среды программирования на изучаемых языках; Уметь: работать в среде программирования; использовать языки программирования высокого уровня; Иметь опыт: проведения тестирования функций отдельных программных и программно-аппаратных средств защиты информации;	
			ПК 2.4.	Знать: типы данных; принципы обработки, хранения и передачи информации ограниченного доступа; Уметь: работать в среде программирования; использовать языки программирования высокого уровня; Иметь опыт: обработки, хранения и передачи информации;	
			ПК 2.6.	Знать: интегрированные среды программирования на изучаемых языках; Уметь: использовать языки программирования высокого уровня; Иметь опыт: регистрации основных событий в автоматизированных (информационных) системах, в том числе с использованием программных и программно-аппаратных средств обнаружения, предупреждения и ликвидации последствий компьютерных атак;	

4	Раздел 4. Модульное программирование	Тема 4.1. Понятие модульного программирования Тема 4.2. Разработка приложений	ОК 01.	Знать: способы решения задач профессиональной деятельности; Уметь: определять способы решения задач профессиональной деятельности;	опрос обучающихся по контрольным вопросам, защита отчетов по лабораторным заданиям, тестирование
			ОК 02.	Знать: типы данных; Уметь: проводить поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности;	
			ОК 03.	Знать: способы планирования и реализации собственного профессионального и личностного развития; Уметь: планировать и реализовывать собственное профессиональное и личностное развитие;	
			ПК 2.1.	Знать: принципы установки и настройки отдельных программных, программно-аппаратных средств защиты информации; Уметь: осуществлять установку и настройку отдельных программных, программно-аппаратных средств защиты информации; Иметь опыт: установки и настройки отдельных программных, программно-аппаратных средств защиты информации;	
			ПК 2.2.	Знать: базовые конструкции изучаемых языков программирования; способы защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами; Уметь: обеспечивать защиту информации в автоматизированных системах отдельными программными, программно-аппаратными средствами; Иметь опыт: обеспечения защиты информации в автоматизированных системах отдельными программными, программно-аппаратными средствами;	
			ПК 2.3.	Знать: базовые конструкции изучаемых языков программирования; интегрированные среды программирования на изучаемых языках; Уметь: работать в среде программирования; использовать языки программирования высокого уровня; Иметь опыт: проведения тестирования функций отдельных программных и программно-аппаратных средств защиты информации;	
			ПК 2.4.	Знать: типы данных; принципы обработки, хранения и передачи информации ограниченного доступа; Уметь: работать в среде программирования; использовать языки программирования высокого уровня; Иметь опыт: обработки, хранения и передачи информации;	
			ПК 2.6.	Знать: интегрированные среды программирования на изучаемых языках; Уметь: использовать языки программирования высокого уровня; Иметь опыт: регистрации основных событий в автоматизированных (информационных) системах, в том числе с использованием программных и программно-аппаратных средств обнаружения, предупреждения и ликвидации последствий компьютерных атак;	

5.2 Типовые контрольные задания или иные материалы

Текущий контроль успеваемости и аттестационные испытания обучающихся могут быть организованы с использованием ресурсов ЭИОС КузГТУ. Полный перечень оценочных материалов расположен в ЭИОС КузГТУ.: <https://el.kuzstu.ru/login/index.php>.

Текущий контроль успеваемости и аттестационные испытания могут проводиться в письменной и (или) устной, и (или) электронной форме.

5.2.1 Оценочные средства при текущем контроле

Текущий контроль по темам дисциплины заключается в опросе обучающихся по контрольным вопросам, защите отчетов по лабораторным и(или) практическим заданиям, тестировании, контрольных работ.

Опрос по контрольным вопросам:

При проведении текущего контроля обучающимся будет письменно, либо устно задано два вопроса, на которые они должны дать ответы.

Например:

1. Перечислите виды алгоритмов
2. Какие существуют способы описания алгоритмов?

Критерии оценивания:

- 90-100 баллов – при правильном и полном ответе на два вопроса;
- 80-89баллов – при правильном и полном ответе на один из вопросов и правильном, но не полном ответе на другой из вопросов;
- 60-79 баллов – при правильном и неполном ответе только на один из вопросов;
- 0-59 баллов – при отсутствии правильных ответов на вопросы.

Количество баллов	0-59	60-79	80-89	90-100
Шкала оценивания	неуд	удовл	хорошо	отлично

Примерный перечень контрольных вопросов:

Раздел 1. Основные принципы алгоритмизации и программирования

Тема 1.1. Основные понятия алгоритмизации

1. Что можно считать алгоритмом?
2. Как называется свойство алгоритма, означающее, что данный алгоритм применим к решению целого класса задач?
3. Какая форма записи алгоритмов обладает наибольшей наглядностью?
4. Как называются величины, значения которых меняются в процессе исполнения алгоритма?
5. Как с помощью знаков математических отношений и логических операторов можно записать выражение $x \in [-10, 10]$?
6. Назовите основные компоненты-блоки алгоритма
7. Перечислите виды алгоритмов
8. Какие существуют способы описания алгоритмов?
9. Какими основными свойствами должен обладать алгоритм?
10. Что такое результат выполнения алгоритма?

Тема 1.2. Принципы разработки алгоритмов

1. Какой алгоритм называется линейным?
2. Какой алгоритм называется цикличным?
3. В каких случаях алгоритм включает в себя ветвление?
4. Какой должен быть список в будущей программе для того, чтобы алгоритм бинарного поиска работал правильно?
5. Может ли компьютер сам указывать на ошибки в алгоритме?
6. Каковы основные принципы разработки алгоритмов?
7. Охарактеризуйте словесно-пошаговый способ записи алгоритмов
8. Какие формы ветвления различают?
9. Какие циклы называют итерационными? Приведите примеры
10. Можно ли по готовой программе составить алгоритм?

Тема 1.3. Языки и системы программирования

1. Какие типы языков программирования существуют?
2. Как называется программная единица, имеющая имя, по которому она может быть вызвана из других частей программы?
3. Какой из наиболее известных языков программирования, используется для обучения программированию в старших классах и на первых курсах вузов?
4. Как называется программное обеспечение, которое служит для создания компьютерных программ на языке программирования?
5. Какие программы в составе системы программирования предназначены для перевода программы из исходного кода в машинные коды?
6. Что определяет язык программирования?
7. На какие группы можно разделить множество языков программирования?
8. Что составляет из символов в процессе программирования?
9. Как влияет удобочитаемость языка программирования на лёгкость создания программ на этом языке?
10. Какое свойство языка программирования даёт возможность более просто переносить программы с одной аппаратурной платформы на другую?

Тема 1.4. Парадигмы программирования

1. Что такое парадигма программирования?
2. Что является инструментом грамматического описания фактов, событий, явлений и процессов в программировании?
3. Какие существуют парадигмы программирования?
4. Какое программирование мало отличается от кодирования?
5. Какие подходы относят к перспективным в программировании?
6. В чем основные принципы парадигмы объектно-ориентированного программирования?
7. Приведите пример языков, поддерживающих процедурную парадигму программирования
8. В чем основные различия между процедурными (императивными) и декларативными языками?
9. Приведите простейший пример директивной программы
10. К какому типу языков можно отнести логическое программирование, основанное на логике предикатов?

Тема 1.5. Принципы отладки и тестового контроля

1. Что понимается под термином Тестирование?
2. Что понимается под термином Отладка?
3. Гарантирует ли прохождение тестирования отсутствие ошибок?
4. Если в процессе отладки и пошагового выполнения не выявлены ошибки, то можно ли сказать, что программа готова к использованию?
5. Что больше влияет на качество готовой программы: тестирование или отладка?
6. Назовите основные этапы процесса тестирования
7. Какие цели преследует тестирование?
8. Возможно ли исчерпывающее тестирование? Почему?
9. Каких правил нужно придерживаться при выполнении отладки программы?
10. Какие существуют методы тестирования?

Раздел 2. Язык программирования

Тема 2.1. Характеристика языка

1. Как называется категория языков программирования, близкая к аппаратному обеспечению компьютера?
2. Является ли язык программирования Java строго-типизированным?
3. К синтаксису каких языков программирования наиболее близок синтаксис C#?
4. Что такое ассемблер?
5. В каких типах языков работа программ рассматривается как последовательное выполнение операторов?
6. Перечислите основные особенности и характеристика языка Pascal
7. Что является критерием отнесения языка программирования к тому или иному уровню?
8. Что означает мощность языка программирования?
9. Что означает надежность языка программирования?
10. Что означает простота языка программирования?

Тема 2.2. Элементы языка. Простые типы данных

1. Какие типы данных относятся к простым?
2. В чем состоит основной признак данных, относящихся к простым типам?
3. Приведите пример текстовой (символьной) информации.
4. Сколько символов содержит основная (исходная американская) таблица кодов ASCII ?
5. В какой системе счисления хранятся целые числа в памяти компьютера?
6. Перечислите основные элементы языка Pascal или Turbo Pascal
7. В каком виде могут изображаться вещественные числа в языке Pascal или Turbo Pascal ?
8. Что такое идентификатор?
9. Что определяет тип данных?
10. Что такое перечислимый и ограниченный типы данных?

Тема 2.3. Базовые конструкции структурного программирования

1. Из каких трёх базовых конструкций состоит любая программа?
2. Сколько входов и выходов должна иметь каждая из трёх базовых управляющих конструкций?
3. Что называется функцией?

4. Как называется алгоритм, который на каком-либо шаге обращается сам к себе?
5. Как называется подпрограмма, которая имеет произвольное количество входных и выходных данных?
6. Какие 3 основные блока можно выделить в программе, написанной на Pascal ?
7. Что называется подпрограммой?
8. Что содержится в разделе операторов в программе на Pascal ?
9. Опишите основные особенности конструкции ветвления
10. Опишите основные особенности конструкции цикла

Тема 2.4. Работа с массивами и указателями. Структурные типы данных

1. Каким командным словом обозначается массив в Pascal ?
2. Какое количество данных может находиться в данном массиве `var ch: array[4..16] of integer;`?
3. Что называется индексом элемента?
4. Каким параметром определяется место элемента в массиве?
5. Что такое массив?
6. Как производится обращение к элементам массива?
7. Какие типы данных могут содержаться в массивах?
8. Какие типы данных называют структурными?
9. Из чего состоит комбинированный структурный тип данных?
10. Как задается структурный тип данных в языке Си (C) ?

Тема 2.5. Процедуры и функции

1. Приведите пример операторов функций
2. Приведите пример операторов процедур
3. Что может возвращать функция в основную программу?
4. Какой вид подпрограммы применяют, если в результате получается несколько величин?
5. Опишите структуру процедуры.
6. Опишите структуру функции.
7. Дайте определения понятиям: формальные значения, фактические значения
8. Опишите механизм передачи параметров из основной программы в подпрограмму и наоборот.
9. Почему при обращении к процедуре, аргумент, передаваемый параметру-переменной, может быть только переменной, а не константой или выражением?
10. Дайте определения понятиям: глобальные переменные, локальные переменные

Тема 2.6. Работа с файлами

1. Как объявляются текстовые файлы в программе?
2. Какие файловые типы данных поддерживает Паскаль?
3. Что нужно обязательно описать при работе с файлами ?
4. Каким оператором файл открывается для чтения?
5. Каким образом записывается условие "конец файла" ?
6. Дайте определение «файл» с точки зрения Pascal
7. Что называют файловым типом переменной?
8. Какова организация доступа к файлам?
9. Назовите основные типы операций работы с файлами
10. Перечислите специальные операции для работы с файлами.

Раздел 3. Основы объектно-ориентированного программирования

Тема 3.1. Класс - как механизм создания объектов

1. Что такое «класс» с точки зрения объектно-ориентированного программирования?
2. Что может быть членом класса ?
3. Приведите пример модификаторов, являющихся модификаторами доступа для класса
4. В каком случае два класса являются дружественными?
5. Для чего используются и что определяют поля, описанные в классе?
6. Что определяют методы, описанные в классе?
7. Что такое экземпляр класса?
8. Согласно какому принципу объектно-ориентированного программирования объединяются поля и методы внутри класса?

9. Что такое «суперкласс»?
10. Какое понятие является более общим – класс или объект?

Тема 3.2. Принципы наследования и полиморфизма

1. Как называется своеобразное проявление полиморфизма, когда два или более различных методов имеют одно и то же имя и различаются только количеством или типами аргументов?
2. В чем заключается принцип полиморфизма?
3. Какие виды наследования бывают?
4. Как называется наследование, при котором класс наследуется более чем от одного класса?
5. Назовите ограничения в наследовании
6. Что называется наследованием?
7. За счет чего реализуется полиморфизм в объектно-ориентированном программировании?
8. Что называется полиморфизмом?
9. Для каких классов наследование реализует механизмы расширения?
10. В каких формах возможно сокрытие имени члена базового класса в производном классе ?

Тема 3.3. Понятия деструктора и конструктора

1. Что называется деструктором?
2. Что называется конструктором?
3. Какой метод может быть конструктором для класса String в языке C ++ ?
4. Какими свойствами обладает конструктор?
5. Сколько конструкторов может быть у класса?
6. По какому признаку в тексте программы можно найти конструктор?
7. Когда вызывается конструктор? Может ли он быть вызван явно?
8. Сколько деструкторов может быть у класса?
9. Когда вызывается деструктор? Может ли он быть вызван явно?
10. Вызывается ли деструктор, если из области видимости выходит динамический объект, адресуемый указателем?

Раздел 4. Модульное программирование

Тема 4.1. Понятие модульного программирования

1. В чем сущность модульного программирования?
2. Можно ли сочетать модульное и структурное программирование?
3. Недостаток модульного программирования?
4. При модульном программировании желательно чтобы какой сравнительный размер имел модуль ?
5. При каком подходе программирования применимо модульное программирование?
6. Как можно использовать модульную технологию программирования в ООП?
7. В чем основное отличие модульного программирования от процедурного?
8. Что такое объектный модуль?
9. Для чего используется стандартный модуль CRT в среде Turbo – Pascal ?
10. Перечислите основные характеристики модуля применительно к Pascal

Тема 4.2. Разработка приложений

1. Какой этап занимает наибольшее время при разработке приложения?
2. Какие стадии входят в разработку приложения ?
3. На каком этапе разработки приложения дается характеристика области применения программы?
4. Зависит ли трудоемкость разработки приложения от сложности алгоритма?
5. Сколько стадий в процессе разработки приложения?
6. С позиции специфики разработки и вида программного обеспечения, на какие два класса делятся задачи?
7. Что представляет собой постановка задачи?
8. Как называется разбиение процесса обработки информации при разработке программ на более простые этапы?
9. Кто осуществляет разработку и отладку программ для решения функциональных задач?
10. Что такое жизненный цикл программного обеспечения?

Отчеты по лабораторным и (или) практическим заданиям(далее вместе - задания):

По каждой работе обучающиеся самостоятельно оформляют отчеты в электронном формате
Содержание отчета:

1. Тема работы.
2. Задачи задания.
3. Краткое описание хода выполнения.
4. Ответы на задания или полученные результаты по окончании выполнения работы (в зависимости от задач, поставленных в п. 2).
5. Выводы

Критерии оценивания:

- 60 – 100 баллов – при раскрытии всех разделов в полном объеме

- 0 – 59 баллов – при раскрытии не всех разделов, либо при оформлении разделов в неполном объеме.

Количество баллов	0-59	60-100
Шкала оценивания	Не зачтено	Зачтено

Процедура защиты отчетов по заданиям:

Оценочными средствами для текущего контроля по защите отчетов являются контрольные вопросы. Обучающимся будет устно задано два вопроса, на которые они должны дать ответы.

Например:

1. Зависит ли трудоемкость разработки приложения от сложности алгоритма?
2. Сколько стадий в процессе разработки приложения?

Критерии оценивания:

- 90-100 баллов – при правильном и полном ответе на два вопроса;

- 80-89 баллов – при правильном и полном ответе на один из вопросов и правильном, но не полном ответе на другой из вопросов;

- 60-79 баллов – при правильном и неполном ответе только на один из вопросов;

- 0-59 баллов – при отсутствии правильных ответов на вопросы.

Количество баллов	0-59	60-79	80-89	90-100
Шкала оценивания	неуд	удовл	хорошо	отлично

Примерный перечень заданий:

1. Задание к лабораторному занятию 1.2.1. Разработка линейных алгоритмов и алгоритмов ветвления.

1. Составьте блок-схему линейного алгоритма для вычисления значения выражения $y = x^3 - 2ax - 10$
2. Составить блок-схему алгоритма ветвления по словесному описанию:

Ввести число.

Если оно неотрицательно, вычесть из него 10, в противном случае прибавить к нему 10.

Вывести результат расчета

Результаты зафиксировать в отчете.

2. Задание к лабораторному занятию 1.2.2. Разработка циклических алгоритмов.

Составить блок-схему циклического алгоритма, который в столбик выводит все целые числа от 20 до 35 и вычисляет корни этих чисел.

Результаты зафиксировать в отчете.

3. Задание к лабораторному занятию 1.2.3. Разработка алгоритмов шифрования

1. Составить блок схему алгоритма, с помощью которого можно зашифровать сообщение по методу Цезаря. Исходное сообщение: «ИНФОРМАЦИЯ». Ключ: 3

Алфавит:

Буква	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й
Номер	1	2	3	4	5	6	7	8	9	10	11
Буква	К	Л	М	Н	О	П	Р	С	Т	У	Ф
Номер	12	13	14	15	16	17	18	19	20	21	22
Буква	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я

Номер	23	24	25	26	27	28	29	30	31	32	33
-------	----	----	----	----	----	----	----	----	----	----	----

2. Составить блок схему алгоритма, с помощью которого можно зашифровать сообщение по методу квадрата Полибия. Исходное сообщение: Жить – родине служить. Условимся, что первая цифра обозначает строку, вторая – столбец.

	1	2	3	4	5	6
1	А	Б	В	Г	Д	Е
2	Ё	Ж	З	И	Й	К
3	Л	М	Н	О	П	Р
4	С	Т	У	Ф	Х	Ц
5	Ч	Ш	Щ	Ъ	Ы	Ь
6	Э	Ю	Я			

Результаты зафиксировать в отчете.

4. Задание к лабораторному занятию 1.4.1. Подготовка конспекта по теме «Типы приложений»

Используя материалы лекции, а также ресурсы сети Интернет, подготовить конспект по теме «Типы приложений».

В конспекте должна быть соблюдена логическая последовательность материала по принципу от общего к частному с поясняющими примерами.

В состав конспекта должно входить как минимум:

- понятие приложения
- признаки для классификации приложений
- среды и основные принципы разработки приложений
- методы оптимизации и тестирования приложений на отсутствие ошибок
- выводы

Результаты зафиксировать в отчете.

5. Задание к лабораторному занятию 2.2.1. Знакомство с инструментальной средой программирования

1. Знакомство с интерфейсом программы:

Найдите строку Меню (сверху) и строку-подсказку (снизу).

Поочередно войдите в указанные ниже разделы Меню и ознакомьтесь с набором команд

2. Набрать простейшую программу, соответствующую условию задачи:

Ввести с клавиатуры два целых числа, найти их сумму, результат вывести на экран с поясняющим текстом.

Внимание! Две косые черты (//) отделяют комментарии, их набирать не нужно.

```
program raschet; // название программы
uses crt; // подключаемые модули
var x, y, s:integer; // объявление имен переменных и их типа
begin // начало исполнительной части
  writeln('Введите два целых числа'); // написать на экране текст
  readln(x,y); // прочитать данные с клавиатуры и запомнить их в переменных
  s:=x+y; // выполнить расчет и запомнить его в переменной
  writeln('Сумма чисел =',s); // вывести на экран текст и значение переменной
end. // конец программы
```

3. Просмотрите текст файла, обратите внимание на структуру программы.

4. Запустите набранную программу на выполнение. Если после запуска программы внизу окна появляется красная строчка с сообщением, внимательно просмотрите всю строчку, найдите и исправьте ошибку.

5. После исправления всех ошибок и появления в новом окне начала работы программы, введите нужные данные (если в программе подразумевается ввод нескольких переменных, то это следует делать через Enter или пробел!), получите результат работы и проверьте его на правильность.

6. Сохраните набранную программу в своей папке.

7. Разберитесь с работой программы и измените ее так, чтобы она вычисляла не сумму, а разность чисел. Проверьте правильность работы измененной программы. Сохраните программу под новым именем в своей папке.

Результаты зафиксировать в отчете.

6. Задание к лабораторному занятию 2.3.1. Разработка программ разветвляющейся структуры.

Задание 1

Даны три действительные числа. Возвести в квадрат те из них, значения которых положительны, и в четвертую степень — отрицательные.

Задание 2

Написать программу, которая вычисляет частное от деления двух чисел. Программа должна проверять правильность введенных пользователем данных и, если они неверные (делитель равен нулю), выдавать сообщение об ошибке. Ниже представлен рекомендуемый вид экрана во время работы программы (данные, введенные пользователем, выделены полужирным шрифтом).

Вычисление частного.

Введите в одной строке делимое и делитель, затем нажмите <Enter>

*-> **12 0***

Вы ошиблись. Делитель не должен быть равен нулю.

Задание 3

Оформить отчет. Отчет должен содержать коды программ

7. Задание к лабораторному занятию 2.3.2. Разработка программ с использованием цикла с предусловием.

Составьте программу решения задачи:

Дано натуральное число n.

Вычислить $S=1!+2!+3!+4!+\dots+n!$ ($n>1$).

Откомпилируйте программу.

Проверьте правильность решения задачи на конкретном примере.

Результаты зафиксировать в отчете.

8. Задание к лабораторному занятию 2.3.3. Разработка программ с использованием цикла с постусловием.

Составьте программу решения задачи:

Введите целое число и определите количество цифр в нем.

Откомпилируйте программу.

Проверьте правильность решения задачи.

Результаты зафиксировать в отчете.

9. Задание к лабораторному занятию 2.3.4. Разработка программ с использованием цикла с параметром.

1. Найти сумму всех чисел, которые делятся на 3 и количество всех остальных из ряда натуральных чисел от 5 до 33.

2. Напишите программу, которая выводит значения функции $y=kx+b$ на интервале $[2,8]$ с шагом 1.

3. Найти сумму ста первых чисел натурального ряда.

4. Вычислить значение

$$y=2ax^2-x, \text{ если } a=\frac{3}{x-4}$$

, переменная x вводится с клавиатуры

Результаты зафиксировать в отчете.

10. Задание к лабораторному занятию 2.4.1. Разработка программ с использованием одномерных массивов и указателей.

1. Задан массив из k символов. Преобразовать массив следующим образом: расположить символы в обратном порядке.

2. Задан массив из k чисел. Найти число наиболее часто встречающееся в этом массиве

3. Задан массив из k чисел. Отсортировать элементы массива по возрастанию

Результаты зафиксировать в отчете.

11. Задание к лабораторному занятию 2.4.2. Сортировка одномерных массивов

1. Дана последовательность действительных чисел a_1, a_2, \dots, a_n . Выяснить, будет ли она возрастающей.

2. Дан массив из N действительных чисел. Подсчитать, сколько в нем отрицательных, положительных и нулевых элементов.
3. Даны действительные числа a_1, a_2, \dots, a_n . Поменять местами первый наибольший элемент с последним наименьшим элементом.
4. В заданном одномерном массиве поменять местами соседние элементы, стоящие на четных местах, с элементами, стоящими на нечетных местах.
5. Задана последовательность $\{X_i\}$ из N вещественных чисел. Вычислить последовательность $\{S_i\}$ по формуле:

$$S_i = \sqrt{\frac{(X_i - M)^2}{N - 1}},$$

, где M — среднее арифметическое значение последовательности X.

Результаты зафиксировать в отчете.

12. Задание к лабораторному занятию 2.4.3. Разработка программ с использованием двумерных массивов.

1. Дана целая квадратная матрица **n-го** порядка. Определить, является ли она магическим квадратом, т. е. такой матрицей, в которой суммы элементов во всех строках и столбцах одинаковы.
2. Определить, является ли заданная целая квадратная матрица **n-го** порядка симметричной (относительно главной диагонали).

Результаты зафиксировать в отчете.

13. Задание к лабораторному занятию 2.4.4. Сортировка двумерных массивов.

1. Упорядочить по возрастанию элементы каждой строки матрицы размером $n \times m$
2. Найти наибольший и наименьший элементы прямоугольной матрицы и поменять их местами.

Результаты зафиксировать в отчете.

14. Задание к лабораторному занятию 2.4.5. Разработка программ с использованием структур.

1. В магазине сформирован список постоянных клиентов, который включает ФИО, домашний адрес покупателя и размер предоставляемой скидки. Удалить из этого списка все повторяющиеся записи, проверив ФИО и домашний адрес покупателя.

2. Список товаров, имеющихся на складе, включает в себя наименование товара, количество единиц товара, цену единицы и дату поступления товара на склад. Вывести в алфавитном порядке список товаров, хранящихся больше месяца, стоимость которых превышает 100 000 р.

Результаты зафиксировать в отчете.

15. Задание к лабораторному занятию 2.4.6. Разработка программ с использованием строк.

1. Дана строка, состоящая из групп нулей и единиц, каждая группа отделяется от другой одним или несколькими пробелами. Найти количество групп с пятью символами.

2. Дана строка, состоящая из групп нулей и единиц. Найти и вывести на экран самую короткую группу.

Результаты зафиксировать в отчете.

16. Задание к лабораторному занятию 2.5.1. Разработка программ с использованием функций.

1. Написать программу функции, которая вычисляет объем цилиндра. Параметрами функции должны быть радиус и высота цилиндра.

2. Написать программу функции, которая возвращает максимальное из двух целых чисел, полученных в качестве аргумента.

Результаты зафиксировать в отчете.

17. Задание к лабораторному занятию 2.5.2. Разработка программ с использованием рекурсивных функций.

1. Написать программу для вычисления

$$S = \sum_{i=1}^n (i + 1)^2 / i$$

двумя методами. Один метод вычисляет сумму без использования рекурсии, другой - с использованием рекурсии.

2. Подсчитать количество цифр в заданном числе.

3. Найти наибольший общий делитель (НОД) чисел M и N, используя метод Эйлера: если M делится на N, то $\text{НОД}(N, M) = N$, иначе $\text{НОД}(N, M) = \text{НОД}(M \% N, N)$.

Результаты зафиксировать в отчете.

18. Задание к лабораторному занятию 2.6.1. Разработка программ работы со структурированными файлами.

На заводе выпускается несколько наименований деталей. Сведения о деталях включают код детали, количество выпущенных деталей, номер месяца выпуска. Вывести информацию о продукции, выпущенной заданным цехом за последний месяц.

Результаты зафиксировать в отчете.

19. Задание к лабораторному занятию 2.6.2. Разработка программ работы с текстовыми файлами.

Написать программу, вводящую в файл или читающую из файла ведомость студентов, сдавших экзамены. Каждая структура должна содержать фамилию, а также оценки по математике и программированию. вывести список студентов, сдавших экзамен по программированию с оценкой 4, и записать эту информацию в текстовый файл.

Результаты зафиксировать в отчете.

20. Задание к лабораторному занятию 2.6.3. Разработка программ работы с неструктурированными файлами.

Записать в файл последовательного доступа N действительных чисел. Вычислить произведение компонентов файла.

Результаты зафиксировать в отчете.

21. Задание к лабораторному занятию 3.1.1. Организация классов и принцип инкапсуляции.

1. Реализовать метод, который возвращает площадь и объем комнаты
2. Задан вектор D(5). Создать класс `vec` осуществляющий ввод данных с клавиатуры, вывод элементов вектора на консоль, и перестановку первого по порядку элемента с k-тым отрицательным элементом вектора. Ограничение доступа в класс (снаружи можно вызывать только `public` части класса)

Результаты зафиксировать в отчете.

22. Задание к лабораторному занятию 3.1.2. Разработка приложений с использованием классов

Задача. Создать приложение, в котором вводятся данные студента, обрабатываются и результат обработки выводится.

1. Создать в среде разработки и сохранить проект под произвольным именем.
 2. Разработать интерфейс проекта.
 3. Написать программный код.
 4. Реализовать интерфейс и программный код в среде визуальной разработки программ.
 5. Провести тестирование и отладку программы.
 6. Нарисовать интерфейс программы со спецификацией и записать программный код с комментариями в отчете по работе.
 7. Записать несколько вариантов тестирования программы.
 8. Провести тестирование исполняемого файла.
 9. Результаты зафиксировать в отчете.
- Результаты зафиксировать в отчете.

23. Задание к лабораторному занятию 3.2.1. Программная реализация принципов наследования.

1. Создайте базовый класс, классы-наследники (`house`-дом и `office` - офис). Используйте механизм `publish`-наследования.

2. Создайте БД по учету студентов. используйте механизм наследования.

Результаты зафиксировать в отчете.

24. Задание к лабораторному занятию 3.2.2. Программная реализация принципов полиморфизма

1. Создайте класс `Лампочка`. Определите для него методы: включить, выключить, установить яркость.
2. Продемонстрируйте работу.

Результаты зафиксировать в отчете.

25. Задание к лабораторному занятию 3.3.1. Разработка конструкторов и деструкторов

Задание: Создать собственный объект с помощью конструкторов и деструкторов и использовать его в программе.

Вариант №1

Объект - овал

1. Свойства - координаты центра, радиусы, цвет линии;
2. Методы - нарисовать, переместить, удалить

Вариант №2

Объект - ломаная линия (^).

1. Свойства - координаты начала и конца, цвет линии;
2. Методы - нарисовать, переместить, удалить

Вариант №3

Объект - прямоугольный треугольник

1. Свойства - координаты точек, цвет линии;
2. Методы - нарисовать, переместить, удалить

Результаты зафиксировать в отчете.

26. Задание к лабораторному занятию 4.2.1. Разработка многомодульных приложений.

Задача. Разработать и реализовать программу ввода данных, вычисления функций при помощи модулей, вывода результатов.

1. Составить список функций и соответствующих им модулей для реализации в программе.
2. Разработать интерфейс программы ввода, обработки и вывода данных.
3. Написать программный код для каждого модуля.
4. Реализовать интерфейс и программный код в среде визуальной разработки программ.
5. Провести тестирование и отладку программы.
6. Нарисовать интерфейс программы со спецификацией и записать программный код с комментариями в отчете по работе.
7. Записать несколько вариантов тестирования программы.
8. Провести тестирование исполняемого файла.

Результаты зафиксировать в отчете.

Тестирование:

Критерии оценивания при тестировании:

- 90-100 баллов - при правильном и полном ответе на 10 вопроса;
- 80...89 баллов - при правильном ответе на 8-9 вопросов;
- 60...79 баллов - при правильном ответе на 5-7 вопросов;
- 0...59 - при правильном ответе только на 4 вопроса;

Количество баллов	0-59	60-79	80-89	90-100
Шкала оценивания	неуд	удовл	хорошо	отлично

Пример тестовых заданий:

Раздел 1. Основные принципы алгоритмизации и программирования

Тема 1.1. Основные понятия алгоритмизации

1. Алгоритмом можно считать:

описание процесса решения квадратного уравнения
расписание уроков в школе
технический паспорт автомобиля
список класса в журнале

2. Как называется свойство алгоритма, означающее, что данный алгоритм применим к решению целого класса задач?

понятность
определённость
результативность
массовость

3. Наибольшей наглядностью обладает следующая форма записи алгоритмов:

словесная

рекурсивная
графическая
построчная

4. Величины, значения которых меняются в процессе исполнения алгоритма, называются:

постоянными
константами
переменными
табличными

5. Какое логическое выражение истинно, если $x \in [-10, 10]$?

$(x > 10)$ И $(x < -10)$
 $(x > 10)$ ИЛИ $(x < -10)$
 $(x < 10)$ ИЛИ $(x > -10)$
 $(x > -10)$ И $(x < 10)$

Тема 1.2. Принципы разработки алгоритмов

1. Линейным называется алгоритм, если:

его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий
он включает в себя вспомогательный алгоритм
он представим в табличной форме

2. Циклическим называется алгоритм, если:

он представим в табличной форме
ход его выполнения зависит от истинности тех или иных условий
он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий

3. Алгоритм включает в себя ветвление, если:

ход его выполнения зависит от истинности тех или иных условий
он включает в себя вспомогательный алгоритм
он представим в табличной форме

4. Для того, чтобы алгоритм бинарного поиска работал правильно нужно, чтобы список был:

несортированным
выходящим из стека
отсортированным

5. Составление алгоритма должно быть особенно ответственным и тщательным, так как машина не может домысливать и исправлять ошибки. Необходимо выбрать общее название некоторого объекта, способного выполнять определенный набор команд:

робот
исполнитель
компьютер

Тема 1.3. Языки и системы программирования

1. Какие типы языков программирования существуют?:

управленческие
основные
процедурные

2. Как называется программная единица, имеющая имя, по которому она может быть вызвана из других частей программы:

файл
программа

подпрограмма

3. Один из наиболее известных языков программирования, используется для обучения программированию в старших классах и на первых курсах вузов, является основой для ряда других языков:

паскаль
пролог
C++
VBA

4. Как называется программное обеспечение, которое служит для создания компьютерных программ на языке программирования?

Среды разработки
Системы программирования
Текстовые процессоры
Языки программирования

5. Какие программы в составе системы программирования предназначены для перевода программы из исходного кода в машинные коды? Выберите несколько из 4 вариантов ответа:

Отладчик
Текстовый редактор
Интерпретатор
Компилятор

Тема 1.4. Парадигмы программирования

1. Парадигма программирования - это

инструмент грамматического описания фактов, событий, явлений и процессов
процесс осуществления неординарного разбора элементов языка
совокупность фактов, влияющих на относительную контекстность любого языка программирования

2. Инструментом грамматического описания фактов, событий, явлений и процессов в программировании является

императивный регулятор
парадигма программирования
дислокационный фактор

3. К парадигмам программирования обычно относят (выбрать все верные):

парадигмы прикладного программирования
парадигмы теоретического программирования
парадигмы функционального программирования

4. Программирование, мало отличающееся от кодирования, называется:

прикладное программирование
теоретическое программирование
функциональное программирование

5. К перспективным подходам к программированию относят (выбрать все верные):

когнитивный подход
регулятивный подход
адаптационный подход

Тема 1.5. Принципы отладки и тестового контроля

1. Выберите более подходящее определение термина Тестирование:

это испытание, проверка правильности работы программы в целом, либо её составных частей
это процесс поиска и устранения ошибок в программе, производимый по результатам её прогона на компьютере

определение реакции программы на некорректные данные или действия пользователя

2. Выберите наиболее верное утверждение:

тестирование показывает наличие ошибок, но не их отсутствие
тестирование показывает отсутствие ошибок
тестирование показывает наличие или отсутствие ошибок

3. Какие возможности обеспечивает программа-отладчик (выбрать все верные)?

пошаговое исполнение программы
просмотр текущего значения любой переменной или нахождение значения любого выражения
установку в программе "контрольных точек"
генерацию корректных и некорректных данных для ввода в программу

4. Если в процессе отладки и пошагового выполнения не выявлены ошибки, то можно ли сказать, что программа готова к использованию?

да
нет

5. Как влияет тестирование на качество готовой программы:

оно не повышает качество, а лишь указывает на его степень
оно повышает качество
оно не повышает и не показывает степень качества, а показывает только наличие или отсутствие ошибок в данном конкретном тесте.

Раздел 2. Язык программирования

Тема 2.1. Характеристика языка

1. Языки близкие к процессору называются языками....

среднего уровня;
низкого уровня;
высокого уровня;
максимального уровня

2. Является ли язык программирования Java строго-типизированным?

Да
Нет

3. К синтаксису каких языков программирования наиболее близок синтаксис C#?

Фортран и Паскаль
Ruby и Python
C++ и Java

4. Что такое ассемблер?

Низкоуровневый язык программирования
Утилита трансляции программы в объектный код компьютера
Высокоуровневый язык программирования

5. В каких типах языков работа программ рассматривается как последовательное выполнение операторов:

объектно-ориентированных;
логических;
процедурных;
математических.

Тема 2.2. Элементы языка. Простые типы данных

1. Какие типы данных относятся к простым? (выбрать все верные)

Символьный

Строковые
Числовые натуральные
Числовые целые
Числовые вещественные
Логические

2. Основной признак данных, относящихся к простым типам, состоит в том, что...

эти данные можно записать двоичным кодом
размер ячейки памяти для записи значения таких данных не превышает 1 байта;
под одним именем записывается только одно значение данных
данные такого типа не могут иметь больше 256 различных значений.

3. Примером текстовой (символьной) информации может служить:

значение числа "пи": 3,14159265358...
график в учебнике по физике или математике;
формула в учебнике по физике или математике;
аудиокнига.

4. Сколько пунктов содержит основная (исходная американская) таблица кодов ASCII :

128
256
65536
1024.

5. Целые числа хранятся в памяти компьютера, будучи записанными:

в шестнадцатеричной системе счисления
в восьмеричной системе счисления
в нормализованном виде
в двоичной системе счисления
в десятичной системе счисления

Тема 2.3. Базовые конструкции структурного программирования

1. Из каких трёх базовых конструкций состоит любая программа?

Последовательности, реструкции, вспомогательной конструкции
Последовательности, ветвления, цикла
Последовательности, разветвления, реструкции

2. Сколько входов и выходов должна иметь каждая из трёх базовых управляющих конструкций?

Один вход и один выход
Один вход и два выхода
Два входа и два выхода

3. Что называется функцией?

Подпрограмма, которая имеет произвольное количество входных и выходных данных
Подпрограмма, которая имеет единственный результат, записываемый в ячейку памяти, имя которой совпадает с именем подпрограммы
Подпрограмма, которая небольшое количество входных и выходных данных, записываемое в ячейку памяти, имя которой не совпадает с именем подпрограммы

4. Как называется алгоритм, который на каком-либо шаге обращается сам к себе?

рекурсивным
циклическим
вспомогательным

5. Как называется подпрограмма, которая имеет произвольное количество входных и выходных данных?

функция
алгоритм
процедура

Тема 2.4. Работа с массивами и указателями. Структурные типы данных

1. Массив обозначается командным словом:

Array
Araу
Array
Arга

2. Какое количество данных может находиться в данном массиве `var ch: array[4..16] of integer;`?

13 целых чисел;
14 целых чисел;
13 букв;
12 дробных чисел.

3. Индексом элемента называется...

Номер элемента в массиве
Последний элемент в массиве
Значение элемента массива

4. Каким параметром определяется место элемента в массиве?

датой
размером
индексом

5. Охарактеризуйте значение указателя в результате выделения памяти для массива из нулевого или отрицательного числа элементов

будет возвращено значение адреса выделенной области памяти
программа аварийно завершится с ошибкой сегментации
будет возвращено заранее непредсказуемое значение
будет возвращено значение NULL

Тема 2.5. Процедуры и функции

1. В приведенном списке выберите функции:

Abs
Round
Randomize
Random

2. В приведенном списке выберите процедуры:

Abs
Round
Randomize
Random

3. Функция может возвращать в основную программу ...

единственное значение
несколько значений
0 значений
все варианты правильные

4. Какой вид программы применяют, если в результате получается несколько величин

и подпрограмму-функцию и подпрограмму-процедуру
подпрограмму-процедуру

нельзя использовать подпрограмму
подпрограмму-функцию

5. Чем процедура отличается от функции?

Процедура возвращает некоторое значение, а функция может возвращать несколько значений или выполнять какое-либо действие

Функция возвращает некоторое значение, а процедура может возвращать несколько значений или выполнять какое-либо действие

Они ничем принципиально друг от друга не отличаются

Тема 2.6. Работа с файлами

1. Как объявляются текстовые файлы в программе?

char
text
doc
string

2. Какие файловые типы данных поддерживает Паскаль?

типизированные
системные
графические
архивные
нетипизированные
текстовые

3. При работе с файлами обязательно надо описать:

подпрограмму
файловую переменную
массивы
строковую переменную

4. Каким оператором файл открывается для чтения?

close
rewrite
assing
reset

5. Выберите условие "конец файла"

if eof(f) then ...
if eoln(f) then ...
if filepos(f) > 0 then ...
if ioresult = 0 then ...

Раздел 3. Основы объектно-ориентированного программирования

Тема 3.1. Класс - как механизм создания объектов

1. Класс - это:

любой тип данных, определяемый пользователем

тип данных, определяемый пользователем и сочетающий в себе данные и функции их обработки
структура, для которой в программе имеются функции работы с нею

2. Членами класса могут быть

как переменные, так и функции, могут быть объявлены как private и как public
только переменные, объявленные как private
только функции, объявленные как private
только переменные и функции, объявленные как private

только переменные и функции, объявленные как public

3. Какие модификаторы являются модификаторами доступа для класса:

extern
abstract
public
internal
integer

4. Что из перечисленного может быть объявлено как в классе, так и вне класса:

классы
структуры
поля
методы
интерфейсы
события

5. Классы А и В являются дружественными классами, если:

один из классов является клиентом другого класса
один из классов является потомком другого класса
классы А и В являются классами одного проекта.

Тема 3.2. Принципы наследования и полиморфизма

1. Своеобразное проявление полиморфизма, когда два или более различных методов имеют одно и то же имя и различаются только количеством или типами аргументов - это:

перезагрузка методов
перегрузка методов
обновление методов
переполнение методов

2. В чем заключается принцип полиморфизма?

в наличии виртуальных методов
в наличии множественного наследования
в использовании виртуального наследования

3. Какие виды наследования бывают (выберите наиболее полный ответ)?

внешнее, внутреннее, защищающее
общее, внешнее, внутреннее, защищающее
внешнее, внутреннее, защищенное

4. Как называется наследование, при котором класс наследуется более чем от одного класса:

множественное
многоуровневое
составное
сложное

5. Назовите ограничения в наследовании (выбрать все верные варианты):

Увеличивает время и усилия, необходимые для выполнения программы, так как она требует переходов между различными классами
Родительский класс и дочерний класс тесно связаны
Любые изменения в программе потребуют изменений как в родительском, так и в дочернем классе.
Необходима тщательная подготовка реализации, иначе это приведет к неверным результатам

6. Что называется наследованием?

это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы

это механизм переопределения методов базового класса

это механизм, посредством которого производный класс получает все поля базового класса

это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить

7. Полиморфизм в объектно-ориентированном программировании реализуется:

через механизмы перегрузки (функций и операций), виртуальные функции и шаблоны

через механизмы перегрузки (функций и операций) и шаблоны;

через виртуальные функции и шаблоны;

через механизмы перегрузки (функций и операций) и виртуальные функции.

8. Полиморфизм - это:

средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов

средство, позволяющее в одном классе использовать методы с одинаковыми именами

средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий

средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов

9. Наследование реализует механизмы расширения для:

базового класса

классов-наследников

10. Соккрытие имени члена базового класса в производном классе возможно в одной из следующих форм: (выбрать все верные)

константа, поле, свойство или внедренный в класс тип или структура скрывают ВСЕ одноименные члены базового класса;

внедренный в класс или структуру метод скрывает в базовом классе свойства, поля, типы, обозначенные тем же самым идентификатором, а также одноименные методы с той же самой сигнатурой;

объявляемые в производном классе индексы скрывают одноименные индексы в базовом классе с аналогичной сигнатурой

Тема 3.3. Понятия деструктора и конструктора

1. Что называется деструктором?

метод, который уничтожает объект

метод, который удаляет объект

метод, который освобождает память, занимаемую объектом

системная функция, которая освобождает память, занимаемую объектом

2. Что называется конструктором?

метод, имя которого совпадает с именем класса и который вызывается автоматически при создании объекта класса

метод, имя которого совпадает с именем класса и который вызывается автоматически при объявлении класса (до создания объекта класса)

метод, имя которого необязательно совпадает с именем класса и который вызывается при создании объекта класса

метод, имя которого совпадает с именем класса и который необходимо явно вызывать из головной программы при объявлении объекта класса

3. Отметьте правильные утверждения (выбрать все верные)

конструкторы класса не наследуются

конструкторов класса может быть несколько, их синтаксис определяется программистом

конструкторов класса может быть несколько, но их синтаксис должен подчиняться правилам перегрузки функций

конструктор возвращает указатель на объект
конструктор не возвращает значение

4. Какой из перечисленных методов может быть конструктором для класса String в языке C++:

void String ()
String * String ()
String (String & s)

5. Выберите верное утверждение о деструкторе класса в C++:

Деструктор принимает в качестве параметра адрес того объекта, который нужно уничтожить
Деструктор не содержит параметров
Деструктор принимает в качестве параметра указатель this

Раздел 4. Модульное программирование

Тема 4.1. Понятие модульного программирования

1. В чем сущность модульного программирования:

в разбиении программы на отдельные функционально независимые части;
в разбиении программы на отдельные равные части;
в разбиение программы на процедуры и функции;

2. Можно ли сочетать модульное и структурное программирование:

да;
нет.

3. Недостаток модульного программирования:

увеличивает трудоемкость программирования;
усложняет процедуру комплексного тестирования;
снижает быстродействие программы;
не позволяет выполнять оптимизацию программы.

4. При модульном программировании желательно, чтобы модуль имел:

большой размер;
небольшой размер;
фиксированный размер;
любой размер.

5. Модульное программирование применимо при:

проектировании сверху вниз;
проектирование снизу-вверх;

Тема 4.2. Разработка приложений

1. Этап, занимающий наибольшее время, при разработке приложения:

тестирование;
сопровождение;
проектирование;
программирование;
формулировка требований

2. В стадии разработки приложения не входит:

автоматизация программирования;
постановка задачи;
составление спецификаций;
эскизный проект;
тестирование.

3. Этап разработки приложения, на котором дается характеристика области применения

программы:

техническое задание;
эскизный проект;
технический проект;
внедрение;
рабочий проект.

4. Зависит ли трудоемкость разработки приложения от сложности алгоритма:

да;
нет.

5. Количество стадий, разработки приложения

7
6
3
5
4

Контрольная работа.

Целью контрольной работы является проверка качества сформированности обучающимися знаний, умений и практических навыков в соответствии с тематикой занятий по дисциплине

Контрольная работа выполняется обучающимися в рамках объема часов отведенного на лабораторное занятие в соответствии с тематическим планом.

Контрольная работа включает в себя два задания.

Критерии оценивания качества выполнения контрольной работы:

- 60-100 баллов - полностью выполнены 2 задания., полностью выполнено 1 задание, 2 задание выполнено частично

- 0 - 59 баллов - оба задания выполнены частично или не выполнены.

Количество баллов	0-59	60-100
Шкала оценивания	не зачтено	зачтено

Темы контрольных работ:

1. Этапы разработки программ
2. Решение задач на базовые конструкции.

Примерные задания к контрольным работам:

Задание к теме № 1. «Этапы разработки программ»

Разработать техническое задание на разработку приложения на произвольную тему, либо в соответствии с вариантом, выданным преподавателем.

Техническое задание должно содержать следующие разделы:

введение;
наименование и область применения;
основание для разработки;
назначение разработки;
технические требования к программе или программному изделию;
техничко-экономические показатели;
стадии и этапы разработки;
порядок контроля и приемки;
приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять

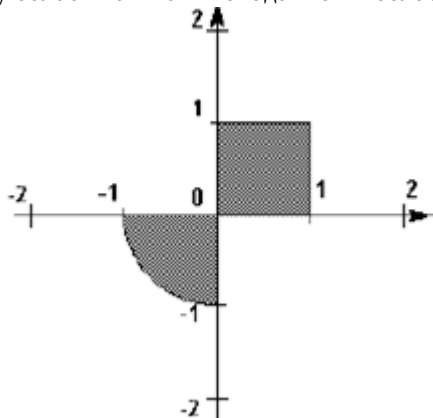
содержание разделов, вводить новые разделы или объединять отдельные из них. При необходимости допускается в техническое задание включать приложения.

Задание к теме № 2. «Решение задач на базовые конструкции»

Вариант 1

1. Вычислить длину окружности, площадь круга и объем шара одного и того же заданного радиуса.

2. Построить программу, которая вводит координаты точки (x, y) и определяет, попадает ли точка в заштрихованную область на рисунке, который соответствует Вашему варианту. Попадание на границу области считать попаданием в область.



3. Вычислить С - наибольший общий делитель натуральных чисел А и В.

4. В одномерном массиве, состоящем из n вещественных элементов, вычислить:

сумму отрицательных элементов массива;

произведение элементов массива, расположенных между максимальным и минимальным элементами.
упорядочить элементы массива по возрастанию.

5. Дана целочисленная прямоугольная матрица. Определить:

количество столбцов, содержащих хотя бы один нулевой элемент;

номер строки, в которой находится самая длинная серия одинаковых элементов

6. Даны несколько строк, содержащих не более 100 символов каждая. Строки состоят из слов, разделенных пробелами. Определить и распечатать самое длинное и самое короткое слово.

7. Составить программу перебора значений величины k от 1 до 30 и вычисления значения функции:

$$y = \begin{cases} \sin x + 2, & \text{при } k = 3 \text{ или } k = 5; \\ \cos x^2, & \text{при } k = 20; \\ \operatorname{tg} x + \sin^2 x, & \text{при } k = 10 \text{ или } k = 15; \end{cases}$$

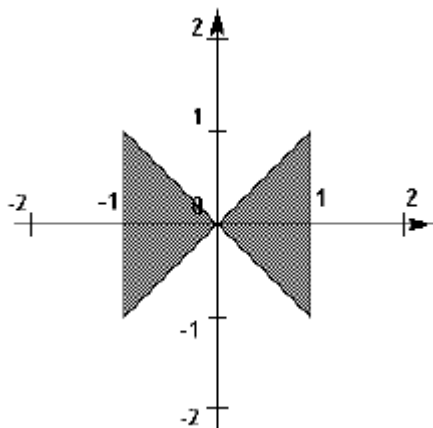
в остальных случаях значение y не определено.

8. Дано 100 вещественных чисел. Определить, образуют ли они возрастающую последовательность.

Вариант 2

1. Вычислить периметр и площадь прямоугольного треугольника по длинам двух катетов.

2. Построить программу, которая вводит координаты точки (x, y) и определяет, попадает ли точка в заштрихованную область на рисунке, который соответствует Вашему варианту. Попадание на границу области считать попаданием в область.



3. Найти U - первый отрицательный член последовательности $\text{Cos}(\text{Ctg } n)$, $n = 1, 2, 3 \dots$.
4. В одномерном массиве, состоящем из n вещественных элементов, вычислить:

произведение элементов массива с четными номерами;
 сумму элементов массива, расположенных между первым и последним нулевыми элементами.
 преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом все отрицательные (элементы, равные 0, считать положительными)

5. Дана целочисленная прямоугольная матрица. Определить:

количество строк, не содержащих ни одного нулевого элемента
 максимальное из чисел, встречающихся в заданной матрице более одного раза.

6. Даны несколько строк, содержащие не более 100 символов каждая. Строки состоят из слов, разделенных пробелами. Вывести на экран эти строки, меняя местами два соседних слова.

7. Дано 100 целых чисел. Определить, сколько из них принимает наибольшее значение.

8. Ввести целое число A в диапазоне от 1 до 5 и любое x .

Если $A=1$, вычислить и напечатать $y = x^3 - x^2 - \sin x$. Если $A = 2$, вычислить и напечатать $y = \cos x - 1,5$.

Если $A=3$, увеличить x в 2 раза и отпечатать.

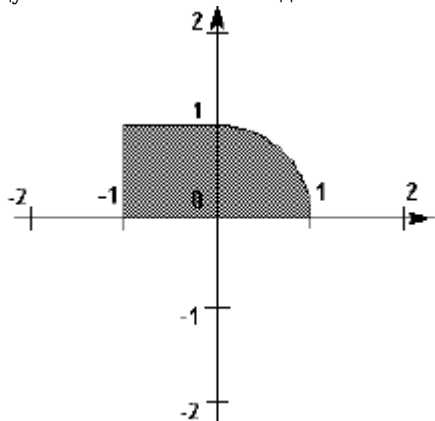
Если $A=4$, вычислить и отпечатать $y = e^{3x} - 1,1x$.

Если $A=5$, выдать сообщение "A=5".

Вариант 3

1. По координатам трех вершин некоторого треугольника найти его площадь и периметр.

2. Построить программу, которая вводит координаты точки (x, y) и определяет, попадает ли точка в заштрихованную область на рисунке, который соответствует Вашему варианту. Попадание на границу области считать попаданием в область.



3. Вычислить $P = (1-1/2^2)(1-1/3^2)\dots(1-1/n^2)$, $n > 2$.

4. В одномерном массиве, состоящем из n вещественных элементов, вычислить:

номер минимального элемента;

сумму элементов массива, расположенных между первым и вторым отрицательными элементами.

преобразовать таким образом, чтобы сначала располагались все элементы, модуль которых не

превышает 1, а потом - все остальные.

5. Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента. Характеристикой строки целочисленной матрицы назовем сумму ее положительных четных элементов. Переставляя строки заданной матрицы, расположить их в соответствии с ростом характеристик.

6. Составить программу вычисления площадей различных геометрических фигур:

$$S = \begin{cases} ab, & \text{при } n = 1; \\ ab / 2, & \text{при } n = 2; \\ (a + b) h / 2, & \text{при } n = 3; \\ \pi r^2, & \text{при } n = 4; \\ \pi r^2 \varphi / 360, & \text{при } n = 5. \end{cases}$$

7. Дано несколько строки, содержащих не более 100 символов каждая. Строки состоят из слов, разделенных пробелами. Определить самое длинное слово в каждой строке и подсчитать количество его вхождений.

8. Дана непустая последовательность целых чисел, среди которых есть два нуля. Найти сумму чисел, расположенных между ними.

5.2.2 Оценочные средства при промежуточной аттестации

Формой промежуточной аттестации в четвертом семестре является экзамен, в процессе которого определяется сформированность обозначенных в рабочей программе компетенций.

Инструментом измерения сформированности компетенций являются:

- зачетные отчеты по заданиям,
- ответы на вопросы во время опроса,
- зачетное компьютерное тестирование.

При проведении промежуточного контроля обучающийся отвечает на 2 вопроса выбранных случайным образом и(или) отвечает на 10 тестовых заданий формирующихся случайным образом, может проводиться в письменной и (или) устной, и (или) электронной форме. Тестирование может проводиться в письменной и (или) устной, и (или) электронной форме. Банк вопросов на тестирование находится в ЭИОС КузГТУ "Moodle".

Ответ на вопросы:

Критерии оценивания при ответе на вопросы:

- 90-100 баллов - при правильном и полном ответе на два вопроса;
- 80-89 баллов - при правильном и полном ответе на один из вопросов и правильном, но не полном ответе на другой из вопросов;
- 60-79 баллов - при правильном и неполном ответе только на один из вопросов;
- 0-59 баллов - при отсутствии правильных ответов на вопросы.

Количество баллов	0-59	60-79	80-89	90-100
Шкала оценивания	неуд	удовл	хорошо	отлично

Примерный перечень вопросов к экзамену:

1. Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма.
2. Принципы структурного программирования. Базовые алгоритмические структуры: следование, ветвление, цикл. Метод пошаговой детализации разработки алгоритмов.
3. Языки программирования. Классификация языков программирования. Понятие уровня языка программирования. Системы программирования.
4. Методы трансляции программ: компиляция, интерпретация.
5. Основные понятия языка программирования: алфавит, синтаксис, семантика. Способы описания языков программирования: языка Бэкуса-Наура формул (БНФ), синтаксические диаграммы Вирта.
6. Система программирования Borland Pascal. Характеристика, назначение. Элементы языка Pascal: алфавит, спец. символы, интерпретация комбинаций некоторых символов, правила использования

- идентификаторов.
7. Типы данных языка Pascal. Классификация, диапазоны значений, размеры.
 8. Операции над данными. Правила записи арифметических выражений. Правила приведения типов в выражениях с разнотипными операндами.
 9. Стандартные функции языка Pascal.
 10. Логические величины. Таблица истинности.
 11. Структура программы на Pascal. Правила описания констант и переменных. Описание собственных типов данных.
 12. Синтаксис основных операторов Pascal: оператор присваивания. Правила использования. Ввод / вывод информации. Форматы вывода.
 13. Программирование ветвящихся алгоритмов. Условный оператор. Примеры.
 14. Оператор варианта CASE. Примеры.
 15. Программирование циклических алгоритмов: цикл с параметром; примеры.
 16. Программирование циклических алгоритмов: цикл с предусловием; примеры.
 17. Программирование циклических алгоритмов: цикл с постусловием; примеры.
 18. Тип данных массив. Описание одномерных и многомерных массивов. Типовые задачи обработки массивов. Примеры.
 19. Методы сортировки массивов.
 20. Строковый тип данных. Операции над строковыми величинами. Процедуры и функции для работы со строками. Примеры.
 21. Подпрограммы. Способы обмена данными между основной программой и подпрограммой: формальные и фактические параметры; параметры-значения и параметры-переменные.
 22. Область действия программных объектов, вложенность подпрограмм, локальные и глобальные описания.
 23. Подпрограммы-процедуры. Примеры использования.
 24. Подпрограммы-функции. Примеры использования.
 25. Рекурсивные подпрограммы. Примеры.
 26. Множества. Операции над множествами. Процедура проверка вхождения элемента в множество. Примеры.
 27. Тип данных "запись". Многоуровневые деревья. Операции с типом запись. Оператор with. Примеры. Примеры работы с массивами записей.
 28. Файлы. Виды файлов. Чтение и запись информации в файл. Процедуры открытия файла для записи, добавления, чтения. Работа с типизированными файлами. Примеры.
 29. Файлы. Организация связи между файловой переменной и внешним файлом. Прямой и последовательный доступ к элементам файла. Работа с текстовыми файлами. Примеры.
 30. Модули. Назначение, структура, примеры использования⁹.

Тестирование:

Критерии оценивания при тестировании:

- 90-100 баллов - при правильном и полном ответе на 10 вопроса;
- 80...89 баллов - при правильном ответе на 8-9 вопросов;
- 60...79 баллов - при правильном ответе на 5-7 вопросов;
- 0...59 - при правильном ответе только на 4 вопроса;

Количество баллов	0-59	60-79	80-89	90-100
Шкала оценивания	неуд	удовл	хорошо	отлично

Пример вариантов тестовых заданий:

Вариант 1.

1. Алгоритмом можно считать:

описание процесса решения квадратного уравнения
 расписание уроков в школе
 технический паспорт автомобиля
 список класса в журнале

2. Как называется свойство алгоритма, означающее, что данный алгоритм применим к решению целого класса задач?

понятность

определённость
результативность
массовость

3. Наибольшей наглядностью обладает следующая форма записи алгоритмов:

словесная
рекурсивная
графическая
построчная

4. Величины, значения которых меняются в процессе исполнения алгоритма, называются:

постоянными
константами
переменными
табличными

5. Какое логическое выражение истинно, если $x \in [-10, 10]$?

$(x > 10) \text{ И } (x < -10)$
 $(x > 10) \text{ ИЛИ } (x < -10)$
 $(x < 10) \text{ ИЛИ } (x > -10)$
 $(x > -10) \text{ И } (x < 10)$

6. Линейным называется алгоритм, если:

его команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий
он включает в себя вспомогательный алгоритм
он представим в табличной форме

7. Циклическим называется алгоритм, если:

он представим в табличной форме
ход его выполнения зависит от истинности тех или иных условий
он составлен так, что его выполнение предполагает многократное повторение одних и тех же действий

8. Алгоритм включает в себя ветвление, если:

ход его выполнения зависит от истинности тех или иных условий
он включает в себя вспомогательный алгоритм
он представим в табличной форме

9. Для того, чтобы алгоритм бинарного поиска работал правильно нужно, чтобы список был:

несортированным
выходящим из стека
отсортированным

10. Составление алгоритма должно быть особенно ответственным и тщательным, так как машина не может домысливать и исправлять ошибки. Необходимо выбрать общее название некоторого объекта, способного выполнять определенный набор команд:

робот
исполнитель
компьютер

Вариант 2.

1. Языки близкие к процессору называются языками....

среднего уровня;
низкого уровня;
высокого уровня;

максимального уровня

2. Является ли язык программирования Java строго-типизированным?

Да
Нет

3. К синтаксису каких языков программирования наиболее близок синтаксис C#?

Фортран и Паскаль
Ruby и Python
C++ и Java

4. Что такое ассемблер?

Низкоуровневый язык программирования
Утилита трансляции программы в объектный код компьютера
Высокоуровневый язык программирования

5. В каких типах языков работа программ рассматривается как последовательное выполнение операторов:

объектно-ориентированных;
логических;
процедурных;
математических.

6. Какие типы данных относятся к простым? (выбрать все верные)

Символьный
Строковые
Числовые натуральные
Числовые целые
Числовые вещественные
Логические

7. Основной признак данных, относящихся к простым типам, состоит в том, что...

эти данные можно записать двоичным кодом
размер ячейки памяти для записи значения таких данных не превышает 1 байта;
под одним именем записывается только одно значение данных
данные такого типа не могут иметь больше 256 различных значений.

8. Примером текстовой (символьной) информации может служить:

значение числа "пи": 3,14159265358...
график в учебнике по физике или математике;
формула в учебнике по физике или математике;
аудиокнига.

9. Сколько пунктов содержит основная (исходная американская) таблица кодов ASCII :

128
256
65536
1024.

10. Целые числа хранятся в памяти компьютера, будучи записанными:

в шестнадцатеричной системе счисления
в восьмеричной системе счисления
в нормализованном виде
в двоичной системе счисления
в десятичной системе счисления

Вариант 3.

1. Своеобразное проявление полиморфизма, когда два или более различных методов имеют одно и то же имя и различаются только количеством или типами аргументов - это:

перезагрузка методов
перегрузка методов
обновление методов
переполнение методов

2. В чем заключается принцип полиморфизма?

в наличии виртуальных методов
в наличии множественного наследования
в использовании виртуального наследования

3. . Какие виды наследования бывают (выберите наиболее полный ответ)?

внешнее, внутреннее, защищающее
общее, внешнее, внутреннее, защищающее
внешнее, внутреннее, защищенное

4. Как называется наследование, при котором класс наследуется более чем от одного класса:

множественное
многоуровневое
составное
сложное

5. Назовите ограничения в наследовании (выбрать все верные варианты):

Увеличивает время и усилия, необходимые для выполнения программы, так как она требует переходов между различными классами
Родительский класс и дочерний класс тесно связаны
Любые изменения в программе потребуют изменений как в родительском, так и в дочернем классе.
Необходима тщательная подготовка реализации, иначе это приведет к неверным результатам

6. Что называется наследованием?

это механизм, посредством которого производный класс получает элементы родительского и может дополнять либо изменять их свойства и методы
это механизм переопределения методов базового класса
это механизм, посредством которого производный класс получает все поля базового класса
это механизм, посредством которого производный класс получает элементы родительского, может их дополнить, но не может переопределить

7. Полиморфизм в объектно-ориентированном программировании реализуется:

через механизмы перегрузки (функций и операций), виртуальные функции и шаблоны
через механизмы перегрузки (функций и операций) и шаблоны;
через виртуальные функции и шаблоны;
через механизмы перегрузки (функций и операций) и виртуальные функции.

8. Полиморфизм - это:

средство, позволяющее использовать одно имя для обозначения действий, общих для родственных классов
средство, позволяющее в одном классе использовать методы с одинаковыми именами
средство, позволяющее в одном классе использовать методы с разными именами для выполнения одинаковых действий
средство, позволяющее перегружать функции для работы с разными типами или разным количеством аргументов

9. Наследование реализует механизмы расширения для:

базового класса
классов-наследников

10. Соккрытие имени члена базового класса в производном классе возможно в одной из следующих форм: (выбрать все верные)

константа, поле, свойство или внедренный в класс тип или структура скрывают ВСЕ одноименные члены базового класса;

внедренный в класс или структуру метод скрывает в базовом классе свойства, поля, типы, обозначенные тем же самым идентификатором, а также одноименные методы с той же самой сигнатурой;

объявляемые в производном классе индексы скрывают одноименные индексы в базовом классе с аналогичной сигнатурой

5.2.3 Методические материалы, определяющие процедуры оценивания знаний, умений, практического опыта деятельности, характеризующие этапы формирования компетенций

Порядок организации проведения текущего контроля и промежуточной аттестации представлен в Положении о проведении текущего контроля и промежуточной аттестации обучающихся, осваивающих образовательные программы среднего профессионального образования в КузГТУ (Ип 06/10

6. Иные сведения и (или) материалы

1. Образовательный процесс осуществляется с использованием как традиционных, так и современных интерактивных технологий. При контактной работе педагогического работника с обучающимися применяются следующие элементы интерактивных технологий:

- совместный разбор проблемных ситуаций;

- совместное выявление причинно-следственных связей вещей и событий, происходящих в повседневной жизни, и их сопоставление с учебным материалом.

2. Проведение групповых и индивидуальных консультаций осуществляется в соответствии с расписанием консультаций по темам, заявленным в рабочей программе дисциплины, в период освоения дисциплины и перед промежуточной аттестацией с учетом результатов текущего контроля.

