

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»
(КГУ)

Кафедра «Автоматизация производственных процессов»



УТВЕРЖДАЮ:

Первый проректор
/ Щербич С.Н. /

«03» сентября 2019 г.

Рабочая программа учебной дисциплины

Программирование и алгоритмизация

образовательной программы высшего образования –
программы бакалавриата

27.03.04 – Управление в технических системах

Направленность:

Системы и технические средства автоматизации и управления

Формы обучения: очная, заочная

Курган 2019

Рабочая программа дисциплины «Программирование и алгоритмизация» составлена в соответствии с учебными планами по программе бакалавриата Управление в технических системах (Системы и технические средства автоматизации и управления) утвержденными :

- для очной формы обучения «29» августа 2019 года;
- для заочной формы обучения «29» августа 2019 года.

Рабочая программа дисциплины одобрена на заседании кафедры «Автоматизация производственных процессов» «02» сентября 2019 года, протокол № 1.

Рабочую программу составил

Старший преподаватель

И.П. Камкин

Согласовано:

Заведующий кафедрой
«Автоматизация производственных
процессов»

Е.К. Карпов

Специалист по учебно-
методической работе
Учебно-методического отдела

Г.В. Казанкова

Начальник Управления
Образовательной деятельности

С.Н. Синицын

1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего 5 зачетных единиц трудоемкости (180 академических часов)

Вид учебной работы	Очная форма	
	На всю дисциплину	Семестр
Аудиторные занятия (всего часов), в том числе:	56	56
Лекции	24	24
Лабораторные работы	32	32
Практические занятия:	-	-
Самостоятельная работа (всего часов), в том числе:	124	124
Курсовая работа (КР)	36	36
Подготовка к экзамену	27	27
Другие виды самостоятельной работы	61	61
Вид промежуточной аттестации (зачет, экзамен)	Э	Э
Общая трудоемкость дисциплины и трудоемкость по семестрам:	180	180

Вид учебной работы	Заочная форма	
	На всю дисциплину	Семестр
Аудиторные занятия (всего часов), в том числе:	12	12
Лекции	4	4
Лабораторные работы	8	8
Практические занятия:	-	-
Самостоятельная работа (всего часов), в том числе:	168	168
Курсовая работа (КР)	36	36
Подготовка к экзамену	27	27
Другие виды самостоятельной работы	105	105
Вид промежуточной аттестации (зачет, экзамен)	Э	Э
Общая трудоемкость дисциплины и трудоемкость по семестрам:	180	180

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Программирование и алгоритмизация» относится к блоку 1 базового модуля учебного плана подготовки.

Изучение дисциплины базируется на результатах обучения, сформированных при изучении следующих дисциплин:

- Математика;
- Информатика;

Результаты обучения по дисциплине необходимы для изучения общеинженерных и специальных технических дисциплин, в частности «Вычислительные машины системы и сети», «Автоматизированные информационно-управляющие системы», «Структуры и алгоритмы обработки данных» и других профилирующих дисциплин, а также в последующей инженерной деятельности; планировании и проведении научно-исследовательских и проектно-конструкторских работ.

Требования к входным знаниям, умениям, навыкам и компетенциям:

- понимание устной (монологической и диалогической) речи на бытовые и общекультурные темы;

- владение навыками поиска, хранения и обработки цифровой информации с использованием средств компьютерной техники;

- способность осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий.

- способность использовать навыки работы с компьютером, владеть методами информационных технологий, соблюдать основные требования информационной безопасности.

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Целью освоения дисциплины «Программирование и алгоритмизация» является: формирование знаний и умений по основам программирования на алгоритмическом языке.

Задачами дисциплины являются: освоение навыков программирования на алгоритмическом языке в интегрированной среде разработки.

Изучение дисциплины направлено на формирование следующих компетенций:

- способностью осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее в требуемом формате с использованием информационных, компьютерных и сетевых технологий (ОПК-6);

- готовностью производить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления (ПК-17).

В результате освоения дисциплины обучающийся должен демонстрировать следующие результаты образования:

1) Знать:

Индекс компетенции(ОК, ПК)	Образовательный результат (указывается формируемые образовательные результаты в рамках соответствующих компетенций)
ОПК-6	Знать: основные методы построения алгоритмов, особенности алгоритмических языков программирования, современные тенденции в развитии языков программирования.
	Знать: методы и средства изучаемого языка программирования, принципы разработки программного обеспечения.
	Знать: Синтаксис и семантику алгоритмического языка программирования.
ПК-17	Знать: принципы работы программного обеспечения
	Знать: Способы использования сторонних библиотек в программном обеспечении собственной разработки

2) Уметь:

Индекс компетенции(ОК, ПК)	Образовательный результат (указывается формируемые образовательные результаты в рамках соответствующих компетенций)
ОПК-6	Уметь: разработать алгоритм решения задачи на компьютере и представить его в графическом виде.
	Уметь: использовать интегрированную среду автоматизированной разработке программ, для получения готового программного продукта.
	Уметь: использовать конструкции языка и стандартные библиотеки для создания приложений
ПК-17	Уметь устанавливать и настраивать интегрированную среду разработки программного обеспечения
	Уметь использовать инструменты автоматизации для разработки программного обеспечения

3) Владеть

Индекс компетенции (ОК, ПК)	Образовательный результат (указывается формируемые образовательные результаты в рамках соответствующих компетенций)
ОПК-6	Владеть: языком программирования.
	Владеть: навыками оформления проектной и конструкторской документацией в соответствии с требованиями ЕСПД.
ПК-17	Владеть: навыками работы на компьютерной технике с пакетами для разработки программных продуктов
	Владеть: навыками настройки программного обеспечения, в том числе самостоятельно разработанного.

4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Учебно-тематический план

Очная форма обучения

Рубеж	Номер раздела, темы	Наименование раздела, темы	Количество часов контактной работы с преподавателем		
			Лекции	Практические занятия	Лабораторные работы
Рубеж 1	1	Алгоритмы, - основные понятия и определения.	1,5	-	-
	2	Алгоритмические языки. Алфавит, синтаксис, семантика.	0,5	-	-
	3	Графический способ описания алгоритмов	1	-	-
	4	Основы языка C++. Переменные, константы, арифметические операции, числовые типы данных, главная функция.	2	-	4
	5	Основы языка C++. Операции. Операторы ветвления, цикл while	3	-	4
	6	Основы языка C++. Символьный тип данных, операторы циклов.	2	-	4
	7	Основы языка C++. Массивы, строки, указатели	2	-	4
		Рубежный контроль №1			2
Рубеж 2	8	Основы языка C++. Функции. Перегрузка функций	2	-	2
	9	Основы языка C++. Пользовательские типы данных. Файлы.	2	-	2
	10	Структурное и	2	-	-

		модульное программирование			
	11	Знакомство с объектно-ориентированным программированием.	4	-	4
	12	Библиотека .NET. Основные классы библиотеки.	2	-	4
		Рубежный контроль №2			2
		Итого:	24		32

Заочная форма обучения

Номер раздела, темы	Наименование раздела, темы	Количество часов контактной работы с преподавателем		
		Лекции	Практические занятия	Лабораторные работы
1	Алгоритмы, - основные понятия и определения.	-	-	-
2	Алгоритмические языки. Алфавит, синтаксис, семантика.	-	-	-
3	Графический способ описания алгоритмов	0,5	-	-
4	Основы языка C++. Переменные, константы, арифметические операции, числовые типы данных, главная функция.	0,5	-	1
5	Основы языка C++. Операции. Операторы ветвления, цикл while	0,5	-	2
6	Основы языка C++. Символьный тип данных, операторы циклов.	0,5	-	2
7	Основы языка C++. Массивы, строки, указатели	1	-	2
8	Основы языка C++. Функции. Перегрузка функций	-	-	-
9	Основы языка C++. Пользовательские типы	-	-	-

	данных. Файлы.			
10	Структурное и модульное программирование	1	-	-
11	Знакомство с объектно-ориентированным программированием.	-	-	1
12	Знакомство с библиотекой .NET. Основные классы библиотеки.	-	-	-
	Итого:	4		8

4.2. Содержание лекций:

Номер раздела, темы	Наименование раздела, темы	Наименование и содержание лекции
1	Алгоритмы, основные понятия и определения.	Цели и задачи курса. Понятия алгоритма, свойства алгоритмов, характеристики алгоритмов. Основные этапы проектирования программных продуктов. Способы описания алгоритмов.
2	Алгоритмические языки. Алфавит, синтаксис, семантика.	Понятие алгоритмического языка, классификация алгоритмических языков, компоненты алгоритмических языков. Понятие об алфавите, синтаксисе и семантике языка программирования.
3	Графический способ описания алгоритмов,	Основные понятия Блок-схемы и начертания графических символов согласно ГОСТ 19.701-90(ИСО 5807-85). Пример Блок-схемы. Алгоритмы линейной, разветвляющейся и циклической структуры. Примеры алгоритмов.
4	Основы языка C++. Переменные, константы, арифметические операции, числовые типы данных, главная функция.	Понятие переменной. Числовые типы данных. Константы. Арифметические операции. Операции инкремента и декремента. Пример реализации консольного приложения на языке C++. Главная функция приложения.
5	Основы языка C++. Операции. Операторы ветвления, цикл	Операции присваивания. Операции сравнения. Логические операции, побитовые логические и сдвиговые операции. Понятие о ветвлении. Конструкция if простая и

	while	<p>составная. Конструкция if – else простая и составная. Каскадный вызов if – else. Конструкция switch-case.</p> <p>Понятие о цикле. Заголовок и тело цикла. Итерация. Оператор while. Пример программы.</p>
6	Основы языка C++. Символьный тип данных, операторы циклов.	<p>Операторы do while, for, break, continue. Оператор goto. Почему не рекомендуется использовать. Символьных тип данных. Кодировка.</p>
7	Основы языка C++. Массивы, строки, указатели	<p>Объявление и использование массивов в языке. Правила использования массивов, массивы символов. Понятие об указателях. Связь массива и указателя. Динамическое распределение памяти. Операторы new и delete.</p>
8	Основы языка C++. Функции. Перегрузка функций	<p>Понятие функции, значение функции в языке C++. Объявление – прототип функции. Передача аргументов в функцию. Время жизни и область действия данных. Примеры определения функции. Понятие о перегрузке функций.</p>
9	Основы языка C++. Пользовательские типы данных. Файлы.	<p>Структуры как составные объекты языка. Перечисления, примеры использования. Файловый ввод-вывод средствами стандартной библиотеки C++.</p>
10	Структурное и модульное программирование	<p>Основы структурного подхода к проектированию программных продуктов. Нисходящее программирование. Понятие модуля программы. Классическая организация модулей в программе на C++. Пример программы.</p>
11	Знакомство с объектно-ориентированным программированием	<p>Особенности объектно-ориентированное программирование. Приемы объектно-ориентированного программирования. Инкапсуляция, наследование единичное и множественное, полиморфизм.</p> <p>Объявление класса. Вложенные классы, конструкторы и деструкторы. Дружественные функции. Перегрузка</p>

		операций и функций. Примеры использования классов.
12	Библиотека .NET. Основные классы библиотеки.	Структура классов библиотеки .NET. Среда CLR. Некоторые классы для разработки оконных приложений Windows. Классы для работы с файлами. Классы для работы с последовательными портами.

4.3. Лабораторные занятия Очная форма обучения

Номер раздела, темы	Наименование раздела, темы	Наименование и содержание лабораторных работ	Трудоемкость, часы
4	Основы языка C++. Переменные, константы, арифметические операции, числовые типы данных, главная функция	Знакомство со средой разработки. Реализация простейшей программы на C++ и форматирование кода. Решение задач на вычисление значений по заданным формулам.	4
5	Основы языка C++. Операции. Операторы ветвления, цикл while	Решение задач на создание программ с ветвящимися алгоритмами.	4
6	Основы языка C++. Символьный тип данных, операторы циклов.	Решение задач на создание программ с циклическими алгоритмами. Символьный вывод кириллического текста.	4
7	Основы языка C++. Массивы, строки, указатели	Решение задач на работу с массивами. Решение задач на работу со строками.	4
		Рубежный контроль 1	2
8	Основы языка C++. Функции. Перегрузка функций	Решение задачи на создание программы состоящей и нескольких функций.	2
9	Основы языка C++. Пользовательские	Решение задачи на создание собственного	2

	типы данных. Файлы.	типа данных, запись и чтение значений этого типа данных в(из) файл(а)	
11	Знакомство с объектно-ориентированным программированием	Создание программ на основе классов.	4
12	Библиотека .NET. Основные классы библиотеки. Приемы разработки программных алгоритмов.	Создание приложения Windows с графическим интерфейсом на основе классов библиотеки .NET. Работа с классами для файлового ввода-вывода, работы с последовательными портами.	4
		Рубежный контроль 2	2
		Итого:	32

Заочная форма обучения

Номер раздела, темы	Наименование раздела, темы	Наименование и содержание лабораторных работ	Трудоемкость, часы
4	Основы языка C++. Переменные, константы, арифметические операции, числовые типы данных, главная функция	Знакомство со средой разработки. Реализация простейшей программы на C++ и форматирование кода. Решение задач на вычисление значений по заданным формулам.	1
5	Основы языка C++. Операции. Операторы ветвления, цикл while	Решение задач на создание программ с ветвящимися алгоритмами.	2
6	Основы языка C++. Символьный тип данных, операторы циклов.	Решение задач на создание программ с циклическими алгоритмами. Символьный вывод	2

		кириллического текста.	
7	Основы языка C++. Массивы, строки, указатели	Решение задач на работу с массивами. Решение задач на работу со строками.	2
11	Знакомство с объектно-ориентированным программированием	Создание программ на основе классов.	1
		Итого:	8

4.4 Курсовая работа.

Курсовая работа является самостоятельной работой студента (две одинаковые курсовые не допускаются).

При выполнении курсовой работы студент должен продемонстрировать навыки программирования, приобретенные при выполнении лабораторных работ и знания, полученные на лекционных занятиях.

Курсовая работа ориентируется на решение задач связанных работой автоматизированных систем управления, разработку приборов для таких систем. Курсовая работа должна содержать:

1. Проект программы, содержащей работоспособный программный код написанный на алгоритмическом языке и откомпилированный в исполняемый модуль, а также необходимые для работы программы данные в виде файлов. (Программа, после установки сделанной в соответствии с описанием, изложенным в прилагаемой записке, должна работать на разных компьютерах отвечающих ограничениям оговоренным в пояснительной записке.)
2. Пояснительную записку, оформленную в соответствии с ГОСТ 7.32
3. Общую блок-схему алгоритма выполняемую на листе А2. Допускается распечатка на 4-х листах А4.

Пояснительная записка должна обязательно содержать:

1. Техническое задание, т.е. конкретизация выбранной темы.
2. Описание общей блок-схемы алгоритма.
3. Описание работы программного продукта с приведением необходимого кода, форм, структуры файлов и т.д..
4. Инструкцию по установке программного продукта, с указанием требований к оборудованию и установленному программному обеспечению.
5. Инструкцию программисту, содержащую всю необходимую информацию об ограничениях и выявленных при тестировании ошибочных ситуациях. Также данная инструкция содержит информацию о возможности модернизации программного продукта.

6. Инструкцию оператору, содержащую порядок действий оператора при рутинной работе с программным продуктом и в случае возникновения нештатных ситуаций.
7. Выводы по проделанной работе.
8. Список используемой литературы.

Примерные темы курсовых работ:

1. Решение задачи сбора информации с нескольких устройств системы «Умный дом» по сетевому протоколу. (Допускается эмуляция устройств с помощью других ПК).
2. Решение систем дифференциальных уравнений численными методами на примере конкретной научной или прикладной задачи.
3. Решение задачи установления связи, обмена данными и командами управления с микроконтроллерным устройством. Разрабатывается приложение для ПК, которое взаимодействует с контроллером.
4. Исследование задачи транспортно-складской операции в автоматизированном производстве.
5. Решение задачи оптимизации технологических процессов в автоматизированном машиностроительном производстве.
7. Решение задачи анализа деятельности предприятия.
8. Автоматизация анализа текстовых документов .
9. Задача организации сбора, хранения и поиска информации на предприятии.
10. Решение частной задачи технического зрения для применения в автоматизации и управлении техническими системами.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

При прослушивании лекций рекомендуется в конспекте отмечать все важные моменты, на которых заостряет внимание преподаватель, в частности те, которые направлены на качественное выполнение соответствующих лабораторных работ.

Преподавателем запланировано использование при чтении лекций и проведении лабораторных занятий технологии учебной дискуссии. Поэтому рекомендуется фиксировать для себя интересные моменты с целью их активного обсуждения на дискуссии в конце лекции и на лабораторных занятиях.

Залогом качественного выполнения лабораторных работ является самостоятельная подготовка к ним путем повторения материалов лекций. Рекомендуется подготовить вопросы по неясным моментам и обсудить их с преподавателем в начале лабораторной работы.

Преподавателем запланировано применение на лабораторных занятиях технологий коллективного взаимодействия. Лабораторные работы выполняются с использованием программного обеспечения: Microsoft Visual Studio 2010, Microsoft Visual Studio Community 2019.

Преподавателем запланировано применение на практических занятиях технологий развивающейся кооперации, коллективного взаимодействия. Поэтому приветствуется групповой метод выполнения заданий и защиты отчетов.

Для текущего контроля успеваемости по очной форме обучения преподавателем используется балльно - рейтинговая система контроля и оценки академической активности. Поэтому настоятельно рекомендуется тщательно прорабатывать материал дисциплины при самостоятельной работе, участвовать во всех формах обсуждения и взаимодействия, как на лекциях, так и на лабораторных занятиях в целях лучшего освоения материала и получения высокой оценки по результатам освоения дисциплины.

Выполнение самостоятельной работы подразумевает самостоятельное изучение разделов дисциплины, подготовку к лабораторным работам к рубежным контролям (для очной формы обучения) , выполнение курсовой работы, подготовку к экзамену.

Рекомендуемая трудоемкость самостоятельной работы представлена в таблице:

Рекомендуемый режим самостоятельной работы

Очная форма обучения

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. час.	
	Очная форма обучения	Заочная форма обучения
Самостоятельное изучение тем дисциплины:	37	97
Изучение правил составления блок-схем	2	6
Изучение операторов и конструкций языка C++	10	30
Изучение базовых типов данных в языке C++.	5	11
Изучение способов получения типов данных в языке C++	4	10
Изучение работы с файлами в языке C++	6	12
Изучение приемов работы с . Microsoft Visual Studio	2	4

Объектно-ориентированного программирования	4	12
Изучение библиотеки .NET	4	12
Подготовка к лабораторным занятиям	16	8
Подготовка к текущему и рубежному контролю	8	-
Выполнение курсовой работы	36	36
Подготовка к экзамену	27	27
Всего:	124	168

6. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

6.1. Перечень оценочных средств

1. Балльно-рейтинговая система контроля и оценки академической активности студентов в КГУ (для очной формы обучения).
2. Отчеты студентов по лабораторным работам.
3. Банк тестовых заданий к рубежным контролям №1, №2(для очной формы обучения).
4. Примерный перечень вопросов для экзамена.
5. Курсовая работа.

6.2 Балльно-рейтинговая система контроля и оценки академической активности студентов в КГУ

№	Наименование	Содержание					
		Распределение баллов					
1	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы (доводятся до сведения студентов на первом учебном занятии)	Вид учебной работы:	Посещение лекций	Выполнение и защита отчетов по лабораторным работам	Рубежный контроль №1	Рубежный контроль №2	Экзамен
		Балльная оценка:	До 12	До 28	До 15	До 15	До 30
		Примечания:	по 1 баллу за 2 часовую лекцию	До 2 баллов за 2 часовую лабораторную, до 4 за 4 часовую			
		Курсовая работа (2 семестр)					

		Объект оценки:	Качество пояснительной записки	Качество программы	Качество доклада	Ритмичность выполнения	Качество защиты
	Бальная оценка:	До 20	До 20	До 20	До 20	Коэффициент от 0.8 до 1.2	До 40
		Студенту могут быть начислены бонусные баллы за активную работу в лаборатории и своевременную сдачу лабораторных работ - до 10 баллов.					
2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и экзамена (зачета)	91- 100 – «отлично» 74-90 – «хорошо» 61-73 – «удовлетворительно» 60 и менее – «неудовлетворительно»					
3	Критерий допуска к итоговому контролю, возможности получения автоматического зачета (экзаменационной оценки) по дисциплине. Возможность получения бонусных баллов.	К экзамену допускаются студенты, набравшие не менее 50 баллов. Студент должен выполнить все лабораторные работы. Получение автоматической экзаменационной оценки возможно при следующих результатах (Rтек + Rруб): 68 баллов - удовлетворительная оценка. Студентам может быть начислено до 10 бонусных баллов за семестр за активную работу на лабораторных работах.					
4	Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) студентов для получения недостающих баллов в конце семестра	В случае если к промежуточной аттестации студент набрал менее 50 баллов: Студент имеет возможность пересдать задания как текущего, так и рубежного контроля. Также отстающим студентам могут быть назначены дополнительные задания по тематике лабораторных работ. Ликвидация академических задолженностей, возникших из-за разности в учебных планах при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем.					

6.3. Процедура оценивания результатов освоения дисциплины

Рубежные контроли и экзамен проводятся в форме тестирования или устного опроса. Тестирование проводится письменно или с применением информационных систем.

Перед проведением каждого рубежного контроля преподаватель прорабатывает со студентами основной материал соответствующих разделов дисциплины в форме краткой лекции-дискуссии. Варианты задания для рубежного контроля №1 состоят из теста, рубежного контроля 2 состоят из задачи. В тесте для рубежного контроля 1 15 вопросов по 1 баллу за каждый.

На каждый рубежный контроль студенту отводится время не менее 45 минут. По результатам рубежного контроля студент получает до 15 баллов.

Неуспевающие студенты могут выполнить другие варианты (не те, что выполняли на рубежном контроле) тестов и задач рубежного контроля.

Преподаватель оценивает в баллах результаты тестирования каждого студента по количеству правильных ответов и заносит в ведомость учета текущей успеваемости. Задача рубежного контроля 2 оценивается по результатам защиты студентом своего решения.

Экзамен проводится в комбинированной форме. Теоретические навыки оцениваются тестированием – до 15 баллов. В тесте 15 вопросов по 1 баллу за каждый. Для оценки практических навыков студенты решают задачу – до 15 баллов. Баллы за задачу выставляются преподавателем по результатам устного опроса. Время, отводимое студенту на экзамен, составляет 1 астрономический час. Результаты текущего контроля успеваемости и экзамена заносятся преподавателем в экзаменационную ведомость, которая сдается в организационный отдел института в день экзамена, а также выставляются в зачетную книжку студента.

6.4 Примеры оценочных средств для рубежных контролей и экзамена

Примерный тест для рубежного контроля 1:

1. Какое свойство алгоритма соответствует определению : выполнение команд алгоритма последовательно, с точной фиксацией моментов окончания выполнения одной команды и начала выполнения следующей, т. е. алгоритм должен содержать последовательность указаний (команд), каждое из которых приводит к выполнению в исполнителе одного шага (действия)
 - а) понятность;
 - б) дискретность;
 - в) определенность;
 - г) результативность;
 - д) массовость.
2. Блок-схема это способ описания описание алгоритма:
 - а) псевдокодом;
 - б) программным кодом;
 - в) графический;
 - г) табличный.
3. Выберите перечень языков программирования, относящихся к объектно-ориентированным языкам;
 - а) C, Pascal, Quick Basic;

б) C++, Object Pascal, Visual Basic, Java;

в) Pascal, Visual Basic, Java;

4. Правила построения фраз, позволяющие определить, правильно или неправильно написана та или иная фраза на языке программирования это:

а) семантика;

б) алфавит языка;

в) синтаксис языка.

5. Какие идентификаторы являются правильными для языка C++.

а) wiggly, HOT_key, _grabl;

б) don't, F123, rde;

в) wiggly, HOT_key, Лава2;

г) 3wiggly, ras, _grabl;

6. Запись main () означает, что это:

а) моя программа;

б) главная функция;

г) функция вычисления максимального значения.

7. Целочисленные переменные объявляются с описателем:

а) float;

б) double;

в) long double;

г) int.

8. Чему равно Z в результате выполнения следующего кода:

```
Intx=7,y = 2, z;
```

```
z = x % y;
```

а) 3.5;

б) 1;

в) 3;

9. Какое значение примет переменная в результате выполнения следующего кода;

```
int i=2;
```

```
switch (i)
```

```
{ case 1: i += 2;
```

```
case 2: i *= 3;
```

```
case 0: i /= 2;
```

```
case 4: i -= 5;
```

```
default: ;
```

```
}
```

а) 2;

б) 6;

в) 5;

г) -2;

д) 3.

10. Какое значение примет переменная i в результате выполнения следующего кода:

```
int t[5]= {3, 5, 12, 6, 15};
```

```
for (int i=0; t[i]<10 ; i++);
```

а) 10;

б) 2;

в) 3;

д) 15;

г) 12;

11. Укажите какой директивой можно подключить библиотеку для операций ввода-вывода **cin** **cout**?

```
#include <iostream>
```

```
#include <locale>
```

```
#include <string>
```

```
#include <in_out>
```

```
#include «stdafx.h»
```

12. Операция логического или в C++ обозначается

```
||
&&
|
&
~
```

13. Операция побитового логического и в C++ обозначается

```
||
&&
|
&
~
```

14. Какие из представленных операторов выполняются, если пользователь введет 1.4?

```
#include <iostream>
```

```
void main()
```

```
{
```

```
double x1,x2,y1,y2, k;
```

```
cin » k;
```

```
if(k>5)
```

```
x1=pow(14,k-1);
```

```
else x1=pow(10,k);
```

```
}
```

```
x1=pow(14,k-1);
```

```
x1=pow(10,k);
```

оба оператора x1=pow(14,k-1); и x1=pow(10,k);

ни один из операторов x1=pow(14,k-1); и x1=pow(10,k);

15. Укажите верное описание заголовка цикла for(1; 2; 3)

1 — операторы выполняемые до запуска тела цикла 1 раз, 2 — условие продолжения цикла, 3 - «шаг» цикла — операторы выполняемые после каждой итерации цикла

1 — операторы выполняемые до запуска тела цикла 1 раз, 2 — условие завершения цикла, 3 - «шаг» цикла — операторы выполняемые после каждой итерации цикла

1 — условие продолжения цикла, 2 — операторы выполняемые до запуска тела цикла 1 раз, 3 - «шаг» цикла — операторы выполняемые после каждой итерации цикла

1 — «шаг» цикла — операторы выполняемые после каждой итерации цикла, 2 — условие продолжения цикла, 3 - операторы выполняемые до запуска тела цикла 1 раз

Пример задания для рубежного контроля 2

Написать программу выполняющую следующие действия:

Пересчитывает, данные объема сыпучих веществ введенные пользователем из метрической системы, в британскую и обратно:

grill(четверть пинты)=0,142065 л

pint (пинта)=0,56826 л

quart (кварта)=1,13652 л

gallon (галлон)=4,54609 л

bushel (бушель)=36,3686 л.

Указание: для пересчета использовать отдельную собственноручно подготовленную функцию.

Примерный вариант теста на экзамен:

1. Операция индексации в C++ - это ...?

Операция, обозначаемая [] и позволяющая обратиться к элементу массива по его номеру

Конструкция обозначаемая if () и позволяющая выполнять или не выполнять оператор в зависимости от условия

Конструкция обозначаемая { } и позволяющая выполнить несколько операторов, как единое целое

Операция обозначаемая () и позволяющая явно определить приоритет выполнения операций

2. Какое значение примет переменная number после выполнения следующего участка кода:

```
int i = 0;
double number = 1;
for(i = 1; i<10; i++)
    number =number* i;
10
11
9
362880
```

3. Укажите содержимое файл test1.txt после выполнения следующего кода:

```
ofstream out1("test1.txt");
ofstream out2("test2.txt");
int i=0,j=3;
out1 <<"Нечетные числа ";
out2 <<"Четные числа ";
```

```

do
{
out1 << j++ <<"";
++i;
j--;
} while(i<=5);

```

Нечетные числа
Нечетные числа 1 3 5
Нечетные числа 1 2 3 4 5 6
Нечетные числа 0 1 2 3 4 5 6
Нечетные числа 3 3 3 3 3

4. В какой строке объявляется указатель на тип данных символ?

```

int * i;
float f;
char c;
float *f;
int i;
char *c;

```

5. Объявлен массив `int arr1[15]`; Выберите верный участок кода в котором в каждый элемент массива будут записаны числа от 15 до 1?

1)

```
for(int i =0; i<=15; i++){
```

```
arr1[i+1] = i;
```

```
}
```

2)

```
int arr1[15];
```

```
int c = 0;
```

```
for(int i =0; i<=14; i++){
```

```
if(i%2 == 0) {
```

```
arr1[c] = i;
```

```
c++;
```

```
}
```

```
}
```

3)

```
int c=0;
```

```
for(int i =0, j =15; i<15; i++,j--){
```

```
if(c == 1) {
```

```
arr1[i] = j;
```

```
}
```

```
}
```

4)

```
iint j = 15;
for(int i =0; i<15; i++)
{
```

```
arr1[i] = j-i;
}
```

6. Представлен участок кода:

```
double * arr1;
int size =0;
cin >> size;
arr1 = new double[size];
for(int i =size-1; i>=0;i--)
arr1[i]=size-i;
```

укажите содержание массива начиная с 0-го элемента, если пользователь ввел число 5.

невозможно определить, такое выполнения программы приведет к ошибке

5 5 5 5 5

1 2 3 4 5

5 4 3 2 1

5 5 5 5 5

7. Укажите где выполнена перегрузка функций

1)

```
int sum(int a, int b)
{
return a+b;
}
double sum(double a, double b)
{
return a+b;
}
```

2)

```
int sum_int(int a, int b)
{
return a+b;
}
double sum_double(double a, double b)
{
return a+b;
}
```

3)

```
int sum(int a, int b)
{
return a+b;
}
```

4)

```
int lsum(int a, int b)
{
return a+b;
}
```

8. Укажите «классическое» оформление модуля программы в C++

- отдельный файл имя_модуля.cs
- два файла имя_модуля.h имя_модуля.cpp
- отдельный файл имя_модуля.sln
- два файла имя_модуля.h, имя_модуля.suo

9. Какие преимущества дает разделение программы на модули по сравнению с программой с одним файлом исходного кода? Выберите все верные ответы

- Упрощение разработки программы за счет снижения количества кода в каждом файле
- Упрощение разработки командой программистов, каждый отвечает за свой модуль
- Позволяет использовать библиотечные функции
- Упрощение компиляции из исходного кода и сборки программы

10. Укажите в какой строке записаны константы типа float?

- "Привет, мир!", "двенадцать", "88.5"
- 'x', 't', '4'
- 5, 8, 10
- 8.0f, 44.4f, 3.14f
- 9.0, 2.7, 53.4

11. Укажите верное обращение к элементам указателя на объект следующей структуры

```
struct Color
{
char R;
char G;
char B; };
```

Color* Red;

- Red::R=255;
- Red->R=255;
- Red[R] = 255;
- Red.R = 255;

12. Класс в объектно-ориентированном программировании – это...

- 1) шаблон для создания объектов, обеспечивающий начальные значения состояний: инициализация полей-переменных и реализация поведения функций или методов.
- 2) способ описания сущности, определяющий состояние и поведение, зависящее от этого состояния, а также правила для взаимодействия с данной сущностью
- 3) сложный тип данных
- 4) все вышеперечисленное

13. Свойство системы, позволяющее объединить данные и методы, работающие с ними, в классе и скрыть детали реализации от пользователя – это...

Инкапсуляция
Полиморфизм
Наследование
Обособление

14. CLR – это ...

(Control linked ratification) – контроль подтверждения сборки программы.

(Common language runtime) — общезыковая исполняющая среда.

(Conventional last rebuilding) – согласованная перестройка проекта программы перед компиляцией

(Control linked ratification) – набор правил подключения библиотечных модулей по стандарту ISO 9586

15. Класс .Net используемый для работы с последовательным портом компьютера

String

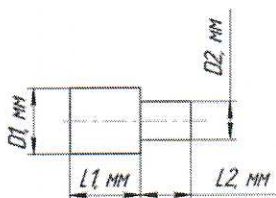
ComboBox

SerialPort

SerialExtended

Пример задачи на экзамен

Написать текст программы нахождения заготовки с минимальной массой из ряда заготовок в виде ступенчатого вала. Известно количество заготовок (задается пользователем). Их линейные размеры и плотность материала задаются случайными числами. Необходимо сделать проверку полученных значений размеров и плотности на формальное соответствие эскизу. Указание: формула для расчета объёма цилиндра – $V=(\pi*d^2)/4$, где d – диаметр основания цилиндра, h – высота



Примерный список вопросов для подготовки к экзамену:

1. Алгоритмы. Основные понятия и определения.
2. Свойства и характеристики алгоритмов.
3. Способы описания алгоритмов.
4. Графическое изображения алгоритмов. Основные изображения блок-схем.
5. Базовые типы данных в C++. Константы в C++. Привести примеры
6. Переменные в C++. Соглашение об именах
7. Арифметические операции в C++. Операции инкремента, декремента. Привести примеры
8. Операции присваивания и модификации. Привести примеры.

9. Операции сравнения. Логические операции. Связь целых и логических значений. Привести примеры.
10. Операции сдвига. Побитовые логические операции. Привести примеры.
11. Оператор if, простой и составной. Привести примеры использования оператора if.
12. Каскадный вызов if — else. Привести примеры
13. Оператор switch, простой и составной. Привести примеры использования оператора switch.
14. Оператор for. Синтаксис, назначение. Привести пример использования
15. Операторы while, do while. Синтаксис, назначение. Привести примеры использования
16. Операторы break и continue. Привести примеры использования.
17. Циклические алгоритмы. Основные понятия. Блок-схема, операторы реализующие циклический алгоритм.
18. Для чего используются массивы? Приведите примеры использования массивов.
19. Для чего используются структуры? Приведите примеры использования структур.
20. Как получить доступ к элементам массива? Как присвоить элементам массива начальные значения? Связь между массивами и указателями.
21. Строки в C и C++. Функции для работы со строками. Привести примеры.
22. Динамическое распределение памяти. Операторы new и delete. Приведите примеры использования
23. Как получить доступ к элементам структуры?
24. Как найти минимальное или максимальное значение в массиве? Приведите примеры.
25. Указатели. Приведите примеры использования указателей.
26. Указатели на массивы и структуры.
27. Массивы указателей. Операции с указателями.
28. Для чего используются файлы? Приведите примеры использования файлов.
29. Как объявить файл? Приведите примеры.
30. Как открыть файл? Приведите примеры.
31. Как записать значения в файл? Приведите примеры.
32. Как организовать форматированное чтение данных из файла?
33. Функции. Для чего используются? Приведите примеры.
34. Функции передача параметров. Приведите примеры.
35. Перегрузка функций. Приведите примеры
36. Модули. Определение, для чего нужны? Привести примеры.
37. Модули. Классическое оформление модуля в C++. Привести пример.
38. Классы, назначение и использование. Методы и свойства классов.
39. Объекты понятие и назначение в языке программирования.
40. Наследование в объектно-ориентированном программировании.
41. Полиморфизм, примеры использования.
42. Инкапсуляция, примеры использования.
43. Привести пример класса, реализованного на C++.
44. В чем разница между структурами и классами?
45. Среда CLR – общее описание. Для чего используется?

46. Библиотека .Net. Что это и для чего используется?
47. Проектирование приложение с графическим интерфейсом на основе .Net.
48. Классы для работы с файлами в .Net.
49. Классы для работы с последовательными портами в .Net.

6.5. Фонд оценочных средств

Полный банк заданий для текущего, рубежных контролей и промежуточной аттестации по дисциплине, показатели, критерии, шкалы оценивания компетенций, методические материалы, определяющие процедуры оценивания образовательных результатов, приведены в учебно-методическом комплексе дисциплины.

7. ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА

7.1 Основная литература

1. Введение в основы программирования на С / Ю.А. Костиков, А.В. Мокряков, В.Ю. Павлов и др. - Москва : НИЦ ИНФРА-М, 2015. - 32 с. ISBN 978-5-16-103253-4. - Текст : электронный. - URL: <https://new.znaniium.com/catalog/product/515180>
2. Немцова, Т. И. Программирование на языке высокого уровня. Программирование на языке С++: учеб. пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 512 с. - ISBN 978-5-16-102802-5. - Текст : электронный. - URL: <https://new.znaniium.com/catalog/product/1000008>

7.2 Дополнительная литература

1. Задачи по программированию / Под ред. Окулов С.М., - 3-е изд. - Москва :Лаборатория знаний, 2017. - 826 с.: ISBN 978-5-00101-448-5. - Текст : электронный. - URL: <https://new.znaniium.com/catalog/product/541059>
2. Хорев, П. Б. Объектно-ориентированное программирование с примерами на С# : учебное пособие / П.Б. Хорев. — Москва : ФОРУМ : ИНФРА-М, 2020. — 200 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-103810-9. - Текст : электронный. - URL: <https://new.znaniium.com/catalog/product/1069921>
3. Страуструп, Б. Язык программирования С++ / Б. Страуструп. – Москва: Бином, 2017. – 1136 с.
4. Давыдов В.Г Программирование и основы алгоритмизации: Учеб. пособие. - М.: Высш.шк , 2003/. -250с.
5. Подбельский В.В. Язык Си++: Учебное пособие. –М.: Финансы и статистика, 2005. -560 с.

8. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

1. Камкин И. П. Методические указания к выполнению курсовой работы по дисциплине «Программирование и алгоритмизация» для студентов очной и заочной форм обучения - Курган, 2019.

2. Камкин И. П. Методические указания к комплексу лабораторных работ по дисциплине «Программирование и алгоритмизация» для студентов очной и заочной форм обучения – Курган, 2019.

9. РЕСУРСЫ СЕТИ «ИНТЕРНЕТ», НЕОБХОДИМЫЕ ДЛЯ ОСВОЕНИЯ ДИСЦИПЛИНЫ

1. <http://www.cetiforum.ru> - Электронные версии учебников, самоучителей по программированию на алгоритмических языках

2. <https://docs.microsoft.com/ru-ru/> - Документация по Microsoft Visual Studio и многое другое.

3. <https://code-live.ru/tag/cpp-manual/> - Простые задачи на C++ для начинающих

4.

https://www.youtube.com/watch?v=SW_UCzFO7X0&list=PLawfWYMUziZqyUL5QDLVbe3j5BKWj42E5&index=2 – Курс CS50 по программированию от Гарварда. Видео лекций на русском языке.

10. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

При чтении лекций используются слайдовые презентации.

Для лабораторных работ необходим программный продукт Microsoft Visual Studio 2010 и Microsoft Visual Studio 2019. Минимальные требования к операционной системе и программному обеспечению компьютера, используемого при чтении лекций: WindowsXP, FoxitReader, Microsoft Visual Studio 2010.

11. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Компьютерный класс, мультимедийное оборудование (переносной персональный компьютер, мультимедийный проектор, мультимедийный экран).

Аннотация к рабочей программе дисциплины
«ПРОГРАММИРОВАНИЕ И АЛГОРИТМИЗАЦИЯ»

образовательной программы высшего образования –
программы бакалавриата

27.03.04 – «Управление в технических системах»

Направленность:

Системы и технические средства автоматизации и управления

Трудоемкость дисциплины: 5 ЗЕ (180 академических часов)

Семестр: 2 (очная форма обучения), 3 (заочная форма обучения)

Форма промежуточной аттестации: Экзамен

Содержание дисциплины

Алгоритмы. Алгоритмические языки. Блок-схемы. Переменные. Основные операции и конструкции языка С++. Массивы. Указатели. Динамическое распределение памяти. Функции. Перегрузка функций.

Структуры как составные объекты языка. Знакомство с объектно-ориентированным программированием. Модульное программирование. Знакомство с библиотекой .NET.