Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курганский государственный университет»
(КГУ)

Кафедра «Автоматизация производственных процессов»

УТВЕРЖДАЮ:
Проректор по образовательной
и международной деятельности
_____ / Кирсанкин А.А. /
«____»_____ 2026 г.

Рабочая программа учебной дисциплины
ПРОГРАММИРОВАНИЕ И АЛГОРИТМИЗАЦИЯ

образовательной программы высшего образования –
программы бакалавриата

15.03.04 – Автоматизация технологических процессов и производств

Направленность:
Автоматизация технологических процессов и производств(в машиностроении)

Формы обучения: очная,заочная

Курган 2026

Рабочая программа дисциплины «Программирование и алгоритмизация» составлена в соответствии с учебными планами по программе бакалавриата «**Автоматизация технологических процессов и производств**» (направленность «**Автоматизация технологических процессов и производств (в машиностроении)**»), утвержденными:

- для очной формы обучения «26» 06 2026 года.
- для заочной формы обучения «26» 06 2026 года.

Рабочая программа дисциплины одобрена на заседании кафедры «Автоматизация производственных процессов» «25» мая 2026 года, протокол № 9.

Рабочую программу составила
доцент И.А.Иванова

Согласовано:

Заведующий кафедрой
«Автоматизация производственных процессов» И.А. Иванова

Специалист по учебно-
методической работе Г.В. Казанкова
Учебно-методического отдела

Начальник Управления
Образовательной деятельности И.В.Григоренко

1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего: 5 зачетных единиц трудоемкости (180 академических часов)

Очная форма обучения

Вид учебной работы	На всю дисциплину	Семестр
		2
Аудиторные занятия (контактная работа с преподавателем), всего часов в том числе:	**56**	**56**
Лекции	24	24
Лабораторные работы	32	32
Самостоятельная работа, всего часов в том числе:	**124**	**124**
Подготовка к экзамену	27	27
Подготовка курсовой работы	36	36
Другие виды самостоятельной работы (самостоятельное изучение тем (разделов) дисциплины)	61	61
Вид промежуточной аттестации	**Экзамен**	**Экзамен**
Общая трудоемкость дисциплины и трудоемкость по семестрам, часов	**180**	**180**

заочная форма обучения

Вид учебной работы	На всю дисциплину	Семестр
		3
Аудиторные занятия (контактная работа с преподавателем), всего часов в том числе:	**12**	**12**
Лекции	4	4
Лабораторные работы	8	8
Самостоятельная работа, всего часов в том числе:	**168**	**168**
Подготовка к экзамену	27	27
Подготовка курсовой работы	36	36
Другие виды самостоятельной работы (самостоятельное изучение тем (разделов) дисциплины)	105	105
Вид промежуточной аттестации	**Экзамен**	**Экзамен**
Общая трудоемкость дисциплины и трудоемкость по семестрам, часов	**180**	**180**

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Программирование и алгоритмизация» относится к блоку 1 обязательным дисциплинам 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 «Структуры и алгоритмы обработки данных», «Вычислительные машины, системы и сети»«Информационные сети и телекоммуникации», «Проектирование автоматизированных систем», «Автоматизированные информационно-управляющие системы» 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;
- знание общих теоретических принципов программирования, методов и средств разработки систем автоматизации и управления;
- владение навыками поиска, хранения и обработки цифровой информации с использованием средств компьютерной техники;
- освоение следующих компетенций на уровне не ниже порогового: ОПК-4 (способность понимать принцип работы современных информационных технологий и использовать их для решения задач профессиональной деятельности) и ОПК-6 (Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий).

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.

Компетенции, формируемые в результате освоения дисциплины:

- Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения(ОПК-14).

- готовность производить инсталляцию и настройку системного, прикладного и инструментального программного обеспечения систем автоматизации и управления (ПК-15).

В результате изучения дисциплины обучающийся должен:

- Знать: основные методы построения алгоритмов, особенности алгоритмических языков программирования, современные тенденции в развитии языков программирования и способы проектирования программного обеспечения (для ПК-15).

- Знать: методы и средства изучаемого языка программирования, особенности интегрированной среды разработки (для ОПК-14).

- Знать: методы и средства программирования в интегрированной среде разработки программ. Синтаксис и семантику алгоритмического языка программирования, принципы и методологию построения алгоритмов (для ПК-15).

- Уметь:разработать алгоритм решения задачи на компьютере и представить его в графическом виде (для ПК-15).

- Уметь:использовать интегрированную среду автоматизированной разработке программ, для получения готового программного продукта (для ОПК-14).

- Уметь:использовать конструкции языка и стандартные библиотеки для создания приложений (для ПК-15).

- Владеть:навыками работы на компьютерной технике с пакетами для разработки программных продуктов, и языком программирования (для ПК-15).

- Владеть:навыками оформления проектной и конструкторской документацией в соответствии с требованиями ЕСКД (для ОПК-14).

Индикаторы и дескрипторы части соответствующей компетенции, формируемой в процессе изучения дисциплины «Программирование и алгоритмизация», оцениваются при помощи оценочных средств.

Планируемые результаты обучения по дисциплине «Программирование и алгоритмизация», индикаторы достижения компетенций ОПК-14,ПК-15, перечень оценочных средств

№ п/п	Код индикатора достижения компетенции	Наименование индикатора достижения компетенции	Код планируемого результата обучения	Планируемые результаты обучения	Наименование оценочных средств
1.	ИД-1о$_{ПК-14}$	Знать: методы и средства изучаемого языка программирования, особенности интегрированной среды разработки	З (ИД-1о$_{ПК-14}$)	Знает: методы и средства изучаемого языка программирования, особенности интегрированной среды разработки	Вопросы для сдачи экзамена
2.	ИД-2о$_{ПК-14}$	Уметь: использовать интегрированную среду автоматизированной разработке	У (ИД-2о$_{ПК-14}$)	Умеет: использовать интегрированную среду автоматизированной разработке	Вопросы для сдачи экзамена

		программ, для получения готового программного продукта		программ, для получения готового программного продукта	
3.	ИД-3о$_{ПК-14}$	Владеть: навыками работы на компьютерной технике с пакетами для разработки программных продуктов, и языком программирования	В (ИД-3о$_{ПК-14}$)	Владеет: навыками работы на компьютерной технике с пакетами для разработки программных продуктов, и языком программирования	Вопросы для сдачи экзамена
4.	ИД-1$_{ПК-15}$	Знать: основные методы построения алгоритмов, особенности алгоритмических языков программирования, современные тенденции в развитии языков программирования и способы проектирования программного обеспечения	З (ИД-1$_{ПК-15}$)	Знает: основные методы построения алгоритмов, особенности алгоритмических языков программирования, современные тенденции в развитии языков программирования и способы проектирования программного обеспечения	Вопросы для сдачи экзамена
5.	ИД-2$_{ПК-15}$	Уметь: использовать интегрированную среду автоматизированной разработке программ, для получения готового программного продукта	У (ИД-2$_{ПК-15)}$	Умеет: использовать интегрированную среду автоматизированной разработке программ, для получения готового программного продукта	Вопросы для сдачи экзамена
6.	ИД-3$_{ПК-15}$	Владеть: навыками оформления проектной и конструкторской документацией в соответствии с требованиями ЕСКД	В (ИД-3$_{ПК-15}$)	Владеет: навыками оформления проектной и конструкторской документацией в соответствии с требованиями ЕСКД	Вопросы для сдачи экзамена

4.СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

4.1. Учебно-тематический план

Очная форма обучения

Рубеж	Номер	Наименование раздела,	Количество часов

	раздела, темы	темы	контактной работы с преподавателем	
			Лекции	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 С++. Переменные, константы, арифметические операции, числовые типы данных, главная функция	2	4
	5	Основы языка С++. Операции. Операторы ветвления, цикл while	3	4
	6	Основы языка С++. Символьный тип данных, операторы циклов.	2	4
	7	Основы языка С++. Массивы, строки, указатели	2	4
		Рубежный контроль 1	-	2
Рубеж 2	8	Основы языка С++. Функции. Перегрузка функций	2	2
	9	Основы языка С++. Пользовательские типы данных. Файлы.	2	2
	10	Структурное и модульное программирование	2	
	11	Знакомство с объектно-ориентированным программированием.	4	4
	12	Библиотека .NET. Основные классы библиотеки.	2	4
		Рубежный контроль 2		2
Всего:			**24**	**32**

заочная форма обучения

	Номер раздела, темы	Наименование раздела, темы	Количество часов контактной работы с преподавателем	
			Лекции	Лабораторные работы
	1	Алгоритмы, - основные понятия и определения.	0,3	-
	2	Алгоритмические языки. Алфавит, синтаксис, семантика.	0,3	-
	3	Графический способ описания алгоритмов	0,3	
	4	Основы языка С++. Переменные, константы, арифметические операции, числовые типы данных, главная функция	0,3	4
	5	Основы языка С++. Операции. Операторы ветвления, цикл while	0,3	
	6	Основы языка С++. Символьный тип	0,3	

		данных, операторы циклов.		
	7	Основы языка С++. Массивы, строки, указатели	0,3	
			-	
	8	Основы языка С++. Функции. Перегрузка функций	0,3	
	9	Основы языка С++. Пользовательские типы данных. Файлы.	0,3	
	10	Структурное и модульное программирование	0,3	
	11	Знакомство с объектно-ориентированным программированием.	0,5	4
	12	Библиотека .NET. Основные классы библиотеки.	0,5	
		Всего:	**4**	**8**

4.2. Содержание лекционных занятий

Тема 1. Алгоритмы, - основные понятия и определения.

Цели и задачи курса. Понятия алгоритма, свойства алгоритмов, характеристики алгоритмов. Основные этапы проектирования программных продуктов. Способы описания алгоритмов.

Тема 2. Алгоритмические языки. Алфавит, синтаксис, семантика.

Понятие алгоритмического языка, классификация алгоритмических языков, компоненты алгоритмических языков. Понятие об алфавите, синтаксисе и семантике языка программирования.

Тема 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.

Тема 4. Основы языка С++. 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 С++. Главная функция приложения.

Тема 5. Основы языка С++. Операции. Операторы ветвления, цикл while.

Операции присваивания. Операции сравнения. Логические операции, побитовые логические операции. Понятие о ветвлении. Конструкция if простая и составная. Конструкция if – else простая и составная. Каскадный вызов if – else. Конструкция switch-case.

Понятие о цикле. Заголовок и тело цикла. Итерация. Оператор while. Пример программы.

Тема 6. Основы языка С++. Символьный тип данных, операторы циклов.

Операторы do while, for, break, continue. Оператор goto. Почему не рекомендуется использовать. Символьных тип данных. Кодировка.

Тема 7. Основы языка С++. Массивы, строки, указатели.

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.

Тема 8. Основы языка С++. Функции. Перегрузка функций.

Понятие функции, значение функции в языке С++. Объявление – прототип функции. Передача аргументов в функцию. Время жизни и область действия данных. Примеры определения функции. Понятие о перегрузке функций.

Тема 9. Основы языка С++. Пользовательские типы данных. Файлы.

Структуры как составные объекты языка. Перечисления, примеры использования. Файловый ввод-вывод средствами стандартной библиотеки С++.

Тема 10. Структурное и модульное программирование.

Основы структурного подхода к проектированию программных продуктов. Нисходящее программирование. Понятие модуля программы. Классическая организация модулей в программе на С++. Пример программы.

Тема 11. Знакомство с объектно-ориентированным программированием.

Особенности объектно-ориентированное программирование. Приемы объектно-ориентированного программирования. Инкапсуляция, наследование единичное и множественное, полиморфизм.

Объявление класса. Вложенные классы, конструкторы и деструкторы. Дружественные функции. Перегрузка операций и функций. Порождение классов. Примеры использования классов. Примеры создания приложений наследуемыми классами.

Тема 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 лабораторной работы	Норматив времени, час.
4	Основы языка С++. Переменные, константы, арифметические операции, числовые типы данных, главная функция.	Знакомство со средой разработки. Реализация простейшей программы на С++ и форматирование кода. Решение задач на вычисление значений по заданным формулам.	4
5	Основы языка С++. Операции. Операторы ветвления, цикл while.	Решение задач на создание программ с ветвящимися алгоритмами.	4

6	Основы языка С++. Символьный тип данных, операторы циклов. Основы языка С++. Массивы, строки, указатели	Решение задач на создание программ с циклическими алгоритмами. Символьный вывод кириллического текста. Решение задач на работу с массивами. Решение задач на работу со строками.	4
7	Основы языка С++. Символьный тип данных, операторы циклов.	Решение задач на создание программ с циклическими алгоритмами. Символьный вывод кириллического текста.	4
		Рубежный контроль 1	2
8	Основы языка С++. Функции. Перегрузка функций	Решение задачи на создание программы состоящей и нескольких функций.	2
9	Основы языка С++. Пользовательские типы данных. Файлы.	Решение задачи на создание собственного типа данных, запись и чтение значений этого типа данных в(из) файл(а)	2
11	Знакомство с объектно-ориентированным программированием.	Создание программ на основе классов. Реализация механизма инкапсуляции, наследования и полиморфизма	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
		Всего:	**32**

заочная форма обучения

Номер раздела, темы	**Наименование раздела, темы**	**Наименование лабораторной работы**	**Норматив времени, час.**
4	Основы языка С++. Переменные, константы, арифметические операции, числовые типы данных, главная функция.	Знакомство со средой разработки. Реализация простейшей программы на С++ и форматирование кода. Решение задач на вычисление значений по заданным формулам.	4
11	Знакомство с объектно-ориентированным программированием.	Создание программ на основе классов. Реализация механизма инкапсуляции, наследования и полиморфизма	4
		Всего:	**8**

4.4. Курсовая работа

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

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

Курсовая работа ориентируется на решение задач, связанных с работой автоматизированный 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 содержать следующие разделы:

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 инсталляции программного продукта с указанием требований к оборудованию и установленному программному обеспечению.
5. Инструкцию программисту, содержащею всю необходимую информацию о 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 использованных источников.

Примерный перечень тем курсовой работы.

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, который взаимодействует с контроллером.
4. Исследование задачи транспортно-складской операции в автоматизированном производстве.
5. Решение задачи оптимизации технологических процессов в автоматизированном машиностроительном производстве.
6. Решение задачи анализа деятельности предприятия.
7. Автоматизация анализа текстовых документов.
8. Задача организации сбора, хранения и поиска информации на предприятии.
9. 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.

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 обучающихся очной формы обучения), выполнение курсовой работы, подготовку к экзамену.

Рекомендуемая трудоемкость самостоятельной работы представлена в таблице:

Рекомендуемый режим самостоятельной работы

Наименование вида самостоятельной работы	**Рекомендуемая трудоемкость, акад. час.**	
	Очная форма обучения	Заочная форма обучения
Самостоятельное изучение тем дисциплины:	**37**	101
Подготовка к лабораторным работам (по 2 часа на каждую лабораторную работу)	16	4
Подготовка к рубежным контролям (по 4 часа на каждый рубеж)	8	-
Изучение правил составления блок-схем	2	12
Изучение операторов и конструкций языка С++	10	12
Изучение базовых типов данных в языке С++.	5	12
Изучение способов получения типов данных в языке С++	4	12
Изучение работы с файлами в языке С++	6	12
Изучение приемов работы с MicrosoftVisualStudio	2	12
Объектно-ориентированное программирование	4	12
Изучение библиотеки .NET	4	17
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 обучающихся в КГУ.
2. Отчеты по лабораторным работам.
3. Банк тестовых заданий к рубежным контролям №1, №2
4. Примерный перечень вопросов к экзамену.
5. Курсовая работа.

6.2. Система балльно-рейтинговой оценки работы обучающихся по дисциплине

Очная форма обучения

№	Наименование	Содержание					
1	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы	Распределение баллов					
		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

<table>
<tr><td></td><td rowspan="5">(доводятся до сведения студентов на первом учебном занятии)</td><td>Примечания:</td><td>по 1 баллу за 2 часовую лекцию</td><td>До 2 баллов за 2 часовую и до 4 баллов за 4 часовую</td><td>На 5-й лабораторной</td><td>На 8-й лабораторной</td><td></td></tr>
<tr><td colspan="6">Курсовая работа (2 семестр)</td></tr>
<tr><td>Объект оценки:</td><td>Качество пояснительной записки</td><td>Качество программы</td><td>Качество доклада</td><td>Ритмичность выполнения</td><td>Качество защиты</td></tr>
<tr><td>Бальная оценка</td><td>До 20</td><td>До 20</td><td>До 20</td><td>Коэффициент от 0,8 до 1,2</td><td>До 40</td></tr>
<tr><td colspan="6">Студенту могут быть начислены бонусные баллы за активную работу в лаборатории и своевременную сдачу лабораторных занятиях - до 10 баллов.</td></tr>
<tr><td>2</td><td colspan="2">Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и экзамена</td><td colspan="5">60 и менее баллов – неудовлетворительно;
61…73 – удовлетворительно;
74… 90 – хорошо;
91…100 – отлично</td></tr>
<tr><td>3</td><td colspan="2">Критерии допуска к промежуточной аттестации, возможности получения автоматического зачета (экзаменационной оценки) по дисциплине, возможность получения бонусных баллов</td><td colspan="5">Для допуска к промежуточной аттестации по дисциплине за семестр (экзамену) обучающийся должен набрать по итогам текущего и рубежного контролей не менее 51 балла .В случае если обучающийся набрал менее 51 балла, то к аттестационным испытаниям он не допускается
Для получения зачета без проведения процедуры промежуточной аттестации обучающемуся необходимо набрать в ходе текущего и рубежных контролей не менее 61 балла. В этом случае итог балльной оценки, получаемой обучающимся без проведения процедуры промежуточной аттестации, определяется по количеству баллов, набранных им в ходе текущего и рубежных контролей. При этом, на усмотрение преподавателя, балльная оценка обучающегося может быть повышена за счет получения дополнительных баллов за академическую активность.
Обучающийся, имеющий право на получение оценки без проведения процедуры промежуточной аттестации, может повысить ее путем сдачи аттестационного испытания. В случае получения обучающимся на аттестационном испытании 0 баллов итог балльной оценки по дисциплине не снижается.
За академическую активность в ходе освоения дисциплины , участие в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности обучающемуся могут быть начислены дополнительные баллы. Максимальное количество дополнительных баллов за академическую активность по одной дисциплине составляет 30.
Основанием для получения дополнительных баллов являются:
- выполнение дополнительных заданий по дисциплине дополнительные баллы начисляются преподавателем;
- участие в течение семестра в учебной, научно-исследовательской, спортивной, культурно-творческой и общественной деятельности КГУ.</td></tr>
</table>

4	Формы и виды учебной работы для неуспевающих (восстановившихся на курсе обучения) обучающихся для получения недостающих баллов в конце семестра	В случае если к промежуточной аттестации (экзамену) набрана сумма менее 51 балла, обучающемуся необходимо набрать недостающее количество баллов за счет выполнения дополнительных заданий, до конца последней (зачетной) недели семестра. Ликвидация академических задолженностей, возникших из-за разности в учебных планах пр переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем

Курсовая работа

Максимальная сумма по курсовой работе устанавливается в 100 баллов.

При оценке качества выполнения работы и уровня защиты рекомендуется следующее распределение баллов:

а) качество пояснительной записки и графической части – до 40 баллов;

б) качество доклада – до 20 баллов;

в) качество защиты работы – до 40 баллов.

При рассмотрении качества пояснительной записки и графической части работы принимается к сведению ритмичность выполнения работы, отсутствие ошибок, логичность и последовательность построения материала, правильность выполнения и полнота расчетов, соблюдение требований к оформлению и аккуратность исполнения работы.

При оценке качества доклада учитывается уровень владения материалом, степень аргументированности, четкости, последовательности и правильности изложения материала, а также соблюдение регламентов.

При оценке уровня качества ответов на вопросы принимается во внимание правильность, полнота и степень ориентированности в материале.

Комиссия по приему защиты курсовой работы (проекта) оценивает вышеуказанные составляющие компоненты и определяет итоговую оценку.

Критерий пересчета баллов за курсовую работу:

60 и менее баллов – неудовлетворительно;

61…73 – удовлетворительно;

74… 90 – хорошо;

91…100 – отлично

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 с обучающимися основной материал соответствующих разделов дисциплины в форме краткой лекции-дискуссии. Варианты заданий для рубежного контроля №1 состоят из теста, рубежного контроля №2 – из одной задачи. В тесте для рубежного контроля №1 15 вопросов по 1 баллу за каждый.

На каждый рубежный контроль обучающемуся отводится время не менее 45 минут. По результатам рубежного контроля обучающийся получает до 15 балов.

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

Преподаватель оценивает в баллах результаты тестирования каждого 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 обучающиеся решают задачу - до 15 баллов. Баллы за задачу выставляются преподавателем по результатам устного опроса. Время, отводимое 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 обучающегося.

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;

```
inti=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:

```
intt[5]= {3, 5, 12, 6, 15};
     for (inti=0; t[i]<10 ;i++) ;
```

а) 10;
б) 2;
в) 3;

г) 12;

д) 15;

11. Укажите какой директивой можно подключить библиотеку для операций ввода-вывода cincout?

а) #include <iostream>

б) #include <locale>

в) #include <string>

г) #include <in_out>

д) #include<stdafx.h>

12. Операция логического ИЛИ в С++ обозначается:

а) ||

б) &&

в) |

г) &

д) ~

13. Операция побитового логического И в С++ обозначается

а) ||

б) &&

в) |

г) &

д) ~

14. Какие из представленных операторов выполнятся, 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);
}
```

а) x1=pow(14,k-1);

б) x1= pow(10,k);

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

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

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

а) 1 –операторы выполняемые до запуска тела цикла 1 раз, 2 – условие продолжения цикла, 3 – «шаг» цикла – операторы выполняемые после каждой итерации цикла;

б) 1 –операторы выполняемые до запуска тела цикла 1 раз, 2 – условие завершения цикла, 3 – «шаг» цикла – операторы выполняемые после каждой итерации цикла;

в) 1 –условие продолжения цикла, 2 – операторы выполняемые до запуска тела цикла 1 раз, 3 – «шаг» цикла – операторы выполняемые после каждой итерации цикла;

г) 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…?

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

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

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

г) Операция, обозначаемая ( ) и позволяющая явно определить приоритет выполнения операций.

2. Какое значение примет переменная numberпосле выполнения следующего участка кода:

```
inti = 0;
double number = 1;
for (i = 1; i<10; i++)
number = number*i;
```

а) 10
б) 11
в) 9
г) 362880

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

```
ofstream out1(“test1.txt”);
ofstream out2(“test2.txt”);
inti = 0, j = 3;
out1 << “Нечетные числа”;
out2<< “Четные числа”;
do
```

```
{
out1 <<j++<< “”;
++i;
j--;
} while (i<=5);
```

а) Нечетные числа

б) Нечетные числа 1 3 5

в) Нечетные числа 1 2 3 4 5 6

г) Нечетные числа 0 1 2 3 4 5 6

д) Нечетные числа 3 3 3 3 3 3

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

а) int*i;

б) float f;

в)charc;

г)float *f;

д)inti;

е) char *c;

5. Объявлен массив intarr1 [15]; Выберите верный участок кода, в котором в каждый элемент массива будут записаны числа от 15 до 1?

а)

```
for (inti = 0; i<= 15; i++){
arr1[i+1] = i;
}
```

б)

```
int arr1[15];
int c = 0;
for (inti = 0; i<= 14; i++){
if(i%2==0) {
arr1[c] = I;
c++;
}
}
```

в)

```
int c = 0;
for (inti = 0; i< 15; i++; j--){
if (c==1) {
arr1[i] = j;
}
}
```

г)

```
int j = 15;
for (inti = 0; i< 15; i++){
arr1[i] = j-1;
}
```

6. Представлен участок кода:

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

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

а) невозможно определить, такое выполнение программы приведет к ошибке

б) 5 5 5 5 5 5

в) 1 2 3 4 5

г) 5 4 3 2 1

д) 5 5 5 5 5

7. Укажите где выполнена перегрузка функций

а)

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

б)

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

в)

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

г)

```
int1sum (int a, int b)
{
return a+b;
}
```

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

а) отдельный файл имя_модуля.cs

б) два файла имя_модуля.cpp

в) отдельный файл имя_модуля.sln

г) 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

а) Упрощение разработки программы за счет снижения количества кода в каждом файле

б) Упрощение разработки командой программистов, каждый отвечает за свой модуль

в) Позволяет использовать библиотечные функции

г) Упрощение компиляции из исходного кода и сборки программы

10. Укажите в какой строке записаны константы типа float?

а) “Привет, мир!”, “двенадцать”, “88.5”

б) ‘x’, ‘t’, ‘4’

в) 5, 8, 10

г) 8.0f, 44.4f, 3.14f

д) 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;
```

а) Red::R = 255;

б) Red->R = 255;

в) Red[R] = 255;

г) Red.R = 255;

12. Класс в объектно-ориентированном программировании – это…

а) шаблон для создания объектов, обеспечивающий начальные значения состояний: инициализация полей-переменных и реализация поведения функций или методов.

б) способ описания сущности, определяющий состояние и поведение, зависящее от этого состояния, а также правила для взаимодействия с данной сущностью.

в) сложный тип данных.

г) все вышеперечисленное.

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

а) Инкапсуляция

б) Полиморфизм

в) Наследование

г) Обособление

14. CLR – это…

а) (Controllinkedratification) – контроль подтверждения сборки программы.

б) (Commonlanguageruntime) – общеязыковая исполняющая среда.

в) (Conventionallastrebuilding) – согласованная перестройка проекта программы перед компиляцией.

г) (Controllinkedratification) – набор правил подключения библиотечных модулей по стандартуISO 9586.

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

а) String

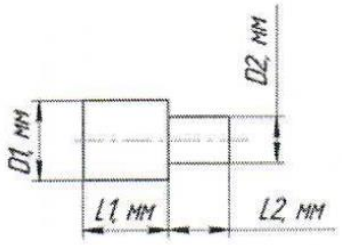
б) ComboBox

в) SerialPort

г) 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 объема цилиндра – V=(pi*d^2)/4, где d – диаметр основания цилиндра, h –высота.



Примерный список вопросов к экзамену

1. Алгоритмы. Основные понятия и определения.
2. Свойства и характеристики алгоритмов.
3. Способы описания алгоритмов.
4. Графическое изображение 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. Операторы 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 примеры использования.
15. Оператор while, dowhile. Синтаксис, назначение. Привести примеры использования.
16. Оператор breaki 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 С и С++. Функции для работы со строками. Привести примеры.
22. Динамическое распределение памяти. Операторы newи delete. Приведите
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 С++. 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 С++.
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 приложения с графическим интерфейсом на основе
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.

7. ОСНОВНАЯ И ДОПОЛНИТЕЛЬНАЯ УЧЕБНАЯ ЛИТЕРАТУРА

7.1. Основная учебная 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.ISBN978-5-16-103253-4. - Текст : электронный. - URL: https://new.znanium.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. — (Среднее профессиональное образование). - ISBN 978-5-16-102802-5. - Текст : электронный. - URL: https://new.znanium.com/catalog/product/1000008

7.2. Дополнительная учебная 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.znanium.com/catalog/product/541059

2. Хорев, П. Б. Объектно-ориентированное программирование с примерами на C# : 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.znanium.com/catalog/product/1069921

3. Давыдов В.Г Программирование и основы алгоритмизации: Учеб. пособие. - М.: Высш.шк , 2003/. -250с.

4. Дорогов, В. Г. Основы программирования на языке С : учеб. пособие / В.Г. Дорогов, Е.Г. Дорогова ; под ред. проф. Л.Г. Гагариной. — Москва : ИД «ФОРУМ» : ИНФРА-М, 2019. — 224 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-16-107656-9. - Текст : электронный. - URL: https://new.znanium.com/catalog/product/1016471

5. Воронцова, Е. А. Программирование на С++ с погружением: практические задания и примеры кода - Москва :НИЦ ИНФРА-М, 2016. - 80

с. ISBN 978-5-16-105159-7. - Текст : электронный. - URL: https://new.znanium.com/catalog/product/563294

6. 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 квыполнению 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 изаочной формы обучения – Курган, 2019.

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

1. dist.kgsu.ru - Система поддержки учебного процесса КГУ;

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

3. http://www.docs.microsoft.com/ru-ru/ - Документация по MicrosoftVisualStudioи многое другое.

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

5. https://www.youtube.com/watch?v=Sy_wba7l1UU&list=PLawfWYMUziZqyUL5QDLVbe3j5BKWj42E5 – Курс CS50 по программированию от Гарварда. Видео лекций на русском языке.

10. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ И ИНФОРМАЦИОННЫЕ СПРАВОЧНЫЕ СИСТЕМЫ

1. ЭБС «Лань»
2. ЭБС «Консультант студента»
3. ЭБС «Znanium.com»
4. «Гарант» - справочно-правовая система

11. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Материально-техническое обеспечение по реализации дисциплины осуществляется в соответствии с требованиями ФГОС ВО по данной образовательной программе.

12. Для студентов, обучающихся с использованием дистанционных образовательных технологий

При использовании электронного обучения и дистанционных образовательных технологий (далее ЭО и ДОТ) занятия полностью или частично проводятся в режиме онлайн. Объем дисциплины и распределение нагрузки по видам работ соответствует п. 4.1. Распределение баллов соответствует п. 6.2, либо может быть изменено в соответствии с решением кафедры, в случае перехода на ЭО и ДОТ в процессе обучения. Решение кафедры об используемых технологиях и системе оценивания достижений обучающихся принимается с учетом мнения ведущего преподавателя и доводится до сведения обучающихся

ПРИЛОЖЕНИЕ 1

Аннотация к рабочей программе дисциплины

«Программирование и алгоритмизация»

образовательной программы высшего образования –
программы бакалавриата

15.03.04 – Автоматизация технологических процессов и производств

Направленность:

Автоматизация технологических процессов и производств (в машиностроении)

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. Алфавит, синтаксис, семантика. Блок-схемы. Переменные. Алгоритмы линейной, разветвляющейся и циклической структуры. Примеры алгоритмов. Основы языка C++. Переменные, константы, арифметические операции, числовые типы данных, главная функция. Операции. Операторы ветвления, цикл while. Символьный тип данных, операторы циклов. Массивы, строки, указатели. Функции. Перегрузка функций. Пользовательские типы данных. Файлы.

Структурное и модульное программирование. Знакомство с объектно-ориентированным программированием. Библиотека .NET. Основные классы библиотеки.

ЛИСТ
регистрации изменений (дополнений) в рабочую программу учебной дисциплины
«Программирование и алгоритмизация»

Изменения / дополнения в рабочую программу на 20___ / 20___ учебный год:

_____
_____
_____
_____
_____
_____

Ответственный преподаватель _____ / Ф.И.О. /

Изменения утверждены на заседании кафедры «___»_____20___ г.,
Протокол № ___

Заведующий кафедрой _____ «___»_____20___ г.

Изменения / дополнения в рабочую программу на 20___ / 20___ учебный год:

_____
_____
_____
_____
_____
_____

Ответственный преподаватель _____ / Ф.И.О. /

Изменения утверждены на заседании кафедры «___»_____20___ г.,
Протокол № ___

Заведующий кафедрой _____ «___»_____20___ г.