

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

Кафедра «Программное обеспечение автоматизированных систем»

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

Рабочая программа учебной дисциплины

Основы программирования

(наименование дисциплины)

образовательной программы высшего образования –
программы бакалавриата

35.03.06 - Агроинженерия

Направленность:
Информационные технологии в электроэнергетике

Форма обучения: очная

Курган 2026

Рабочая программа дисциплины «Основы программирования» составлена в соответствии с учебными планами по программе бакалавриата Агроинженерия (Информационные технологии в электроэнергетике), утвержденным:
- для очной формы обучения «26» июня 2026 года.

Рабочая программа дисциплины одобрена на заседании кафедры «Программное обеспечение автоматизированных систем» «29» июня 2026 года, протокол № 12.

Рабочую программу составил:

ст.преподаватель _____ Ю.В. Адаменко

Согласовано:

Заведующий
кафедрой ПОАС _____ С.В. Косовских

Заведующий кафедрой
«Цифровая энергетика» _____ Ж.В. Нечеухина

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

Начальник управления образовательной
деятельности _____ И.В. Григоренко

1. ОБЪЕМ ДИСЦИПЛИНЫ

Всего: 6 зачетных единиц трудоемкости (216 академических часов)

Очная форма обучения

Вид учебной работы	На всю дисциплину	Семестр	
		3	4
Аудиторные занятия (контактная работа с преподавателем), всего часов	120	48	72
в том числе:			
Лекции	56	32	24
Лабораторные работы	48	16	32
Практические работы	16	-	16
Самостоятельная работа, всего часов	96	60	36
в том числе:			
Подготовка к зачету	36	18	18
Другие виды самостоятельной работы (самостоятельное изучение тем (разделов) дисциплины)	60	42	18
Вид промежуточной аттестации	Зачет	Зачет	Зачет
Общая трудоемкость дисциплины и трудоемкость по семестрам, часов	216	108	108

2. МЕСТО ДИСЦИПЛИНЫ

В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Дисциплина «Основы программирования» относится к дисциплинам обязательной части блока 1.

Изучение дисциплины базируется на знаниях, умениях и навыках, приобретенных студентами в средней школе и на результатах обучения, сформированных при изучении следующих дисциплин: «Информатика» и «Математика».

Результаты обучения по дисциплине необходимы для изучения дисциплин: «Информатика в задачах энергетики», «Методы и алгоритмы анализа данных», «Основы искусственного интеллекта».

3. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Цель освоения дисциплины: изучение современных методов создания качественного программного обеспечения, удовлетворяющего заданным требованиям, знакомство с современными методами разработки простейших программных приложений. Формирование общепрофессиональных и специальных компетентностей посредством знакомства обучающихся с базовыми понятиями программирования; формирование умения анализировать поставленную задачу и на основе анализа выбирать соответствующие средства языка программирования для ее реализации.

Задачи дисциплины: используя ресурсы образовательной программы, университетского образовательного пространства, профессионального сообщества способствовать формированию у обучающихся навыков решения стандартных задач профессиональной деятельности на основе информационной культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; заложить основы программирования приложений и создания программных прототипов решения прикладных задач.

Компетенции, формируемые в результате освоения дисциплины.

- Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности (ОПК-7).

В результате изучения дисциплины обучающийся должен
знать:

- принципы работы современных информационных технологий для решения задач профессиональной деятельности (ОПК-7);

уметь:

- использовать современные информационные технологии для решения задач профессиональной деятельности (ОПК-7)

владеть:

- основами языков программирования и библиотек для разработки приложений и создания программных прототипов для решения прикладных задач (ОПК-7).

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

Планируемые результаты обучения по дисциплине «Основы программирования», индикаторы достижения компетенций ОПК-7 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-1 ОПК-7	Знать: принципы работы современных информационных технологий для решения задач профессиональной деятельности	З (ИД-1 ОПК-7)	Знает: принципы работы современных информационных технологий для решения задач профессиональной деятельности	Отчеты по лабораторным и практическим работам Вопросы для сдачи зачета
2.	ИД-2 ОПК-7	Уметь: использовать современные информационные технологии для решения задач профессиональной деятельности	У (ИД-2 ОПК-7)	Умеет: использовать современные информационные технологии для решения задач профессиональной деятельности	Отчеты по лабораторным и практическим работам Вопросы для сдачи зачета
3.	ИД-3 ОПК-7	Владеть: основами языков программирования и библиотек для разработки приложений и создания программных прототипов для решения прикладных задач	В (ИД-3 ОПК-7)	Владеет: основами языков программирования и библиотек для разработки приложений и создания программных прототипов для решения прикладных задач	Отчеты по лабораторным и практическим работам Вопросы для сдачи зачета

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	Практич. занятия	Лабора- торные работы
I семестр					
Рубеж 1	1.	Переменные. Типы данных. Операторы.	4		1
	2.	Условные и циклические конструкции	2		
		3.	Строки.	4	

	4.	Структуры данных (списки, кортежи, множества, диапазоны)	6		3
	5.	Функции.	4		1
		Рубежный контроль № 1	-		2
Рубеж 2	6.	Регулярные выражения	4		1
	7.	Структуры данных. Словари	2		2
	8.	Работа с файлами и каталогами	4		1
	9.	Работа с изображениями	2		2
		Рубежный контроль № 2			2
Всего			32		16
<i>2 семестр</i>					
Рубеж 3	10.	Общие сведения о программах, лексемах и алфавите.	2		6
	11.	Основные управляющие конструкции C++.	4		
	12.	Указатели и массивы в C++.	2	4	4
	13.	Функции в C++.	2	4	
	14.	Строки и векторы в C++.	6		6
		Рубежный контроль №3		2	
Рубеж 4	15.	Структуры и объединения в C++.	2		8
	16.	Работа с файлами в C++.	2		8
	17.	Обработка исключительных ситуаций в C++.	2	4	
	18.	Основные компоненты, используемые для создания приложений Windows Forms	2		-
		Рубежный контроль № 4		2	
Всего			24	16	32
Итого			56	16	48

4.2. Содержание лекционных занятий

1 семестр

Тема 1 Переменные. Типы данных. Операторы

Присваивание значений переменным. Проверка типа данных. Преобразование типов. Основные операторы: двоичные, работы с последовательностями, присваивания.

Тема 2 Условные и циклические конструкции

Операторы сравнения. Условная конструкция. Циклы. Функции range() и enumerate(). Операторы continue и break.

Тема 3 Строки

Основные операции со строками. Функции и методы для работы со строками: поиск и замена в строке, изменение регистра символов, работа с символами, проверка типа содержимого строки.

Тема 4 Структуры данных (списки, кортежи, множества, диапазоны)

Создание списка. Основные операции над списками. Многомерные списки. Перебор элементов списка. Генераторы списков. Методы работы со списками (добавление и удаление

элементов списка, поиск элемента, получение сведений о значениях, входящих в список, сортировки списка).

Кортежи, их отличия от списков. Множества, особенности определения и использования. Основные операции над множествами. Диапазоны, способы задания и область применения.

Словари. Создание словаря. Операции над словарями. Перебор элементов словаря. Основные методы работы со словарями.

Тема 5 Функции

Описание функции и обращение к ней. Особенности использования функций. Анонимные функции. Декораторы функций. Рекурсия. Глобальные и локальные переменные. Вложенные функции.

Тема 6 Регулярные выражения

Синтаксис регулярных выражений. Поиск первого совпадения с шаблоном. Поиск всех совпадений с шаблоном. Замена в строке.

Тема 7 Структуры данных. Словари

Словари. Создание словаря. Операции над словарями. Перебор элементов словаря. Основные методы работы со словарями.

Тема 8 Работа с файлами и каталогами

Открытие файла. Основные функции и методы работы с файлами. Права доступа к файлам и каталогам. Функции работы с каталогами. Сохранение объектов в файл.

Тема 9 Работа с изображениями

Библиотека Pillow, ее базовые возможности. Загрузка изображений. Создание нового изображения. Получение информации об изображении. Рисование линий и фигур.

2 семестр

Тема 10 Общие сведения о программах, лексемах и алфавите

Лексические основы языка C++. Алфавит. Лексемы. Скалярные типы и выражения. Примеры. Использование простейших типов. Примеры. Функции форматного ввода и вывода.

Операции. Знаки операций. Арифметические операции. Примеры. Инкремент и декремент. Примеры. Операции присваивания и отношения. Примеры. Логические операции. Примеры. Условная операция. Примеры. Операции преобразования типов. Примеры.

Тема 11 Основные управляющие конструкции C++

Условные конструкции. Оператор if...else. Примеры. Оператор switch. Примеры.

Циклические конструкции. Цикл while. Примеры. Цикл do...while. Примеры. Цикл for. Примеры. Оператор безусловного перехода. Примеры. Оператор принудительного выхода из цикла или переключателя. Примеры. Оператор завершения выполнения текущего шага тела цикла. Примеры. Оператор возвращающий значение из функции. Примеры.

Тема 12 Указатели и массивы в C++

Указатели и адреса. Понятие указателя и адреса. Операция косвенной адресации и нахождения адреса. Адресная арифметика. Операция sizeof.

Массивы. Общие сведения о массивах. Одномерные массивы. Имя массива. Двумерные массивы. Многомерные массивы. Работа с массивами с помощью указателей. Примеры. Массивы указателей. Примеры. Массивы динамической памяти. Примеры.

Тема 13 Функции в C++

Функции. Общие сведения о функциях. Определение функции. Аргументы функции. Примеры. Передача нескольких значений в функцию. Возврат нескольких значений из функции. Примеры.

Ссылки. Общие сведения о ссылках. Ссылки в качестве параметров функций. Ссылки в качестве результатов функций. Классы памяти. Рекурсия. Функции с переменным числом параметров. Подставляемые (inline) функции. Способы передачи массивов в функции и их возврата. Примеры.

Тема 14 Строки и векторы в C++

Строки. Строковая константа. Инициализация строк. Строки и указатели. Функции для работы со строками. Использование строковых функций. Использование строк в командной строке. Примеры.

Класс string: назначение и способы задания объектов. Основные операции над объектами класса string. Функции для работы с отдельными символами строки. Основные методы класса string.

Класс vector: назначение и способы задания объектов. Понятие итератора. Арифметические операции над итераторами. Основные методы работы с векторами.

Тема 15 Работа с файлами в C++

Понятие файла. Ввод/вывод файлов в языке C. Основные функции для работы с файлами. Стандартный ввод/вывод. Функции стандартного ввода/вывода в языке C.

Ввод/вывод в языке C++. Пространства имен. Классы fstream, ofstream, ifstream; работа с файлами с использованием этих классов. Бинарные файлы. Стандартный ввод/вывод в C++.

Тема 16 Структуры и объединения в C++

Структуры. Описание структур. Операции над структурами. Использование полей битов в качестве полей структур.

Объединения. Понятие объединения. Совместное использование объединений и структур. Переменные структур.

Тема 17 Обработка исключительных ситуаций в C++

Общие принципы механизма обработки исключений. Синтаксис и семантика генерации и обработки исключений.

Операторы try, catch и throw. Генерация исключений. Основные классы типов исключений. Функции, выдающие исключения.

Тема 18 Основные компоненты, используемые для создания приложений Windows Forms

Понятие CLR-приложения. Описание массивов и структур в CLR-приложениях. Указатели в CLR-приложениях.

Пространство имен System. Основные типы переменных, используемые при создании приложений Windows Forms.

Структура приложения Windows Forms, назначение некоторых файлов.

Понятие компонента, свойства, метода, события. Основные компоненты, используемые при создании приложений Windows Forms (Form, Button, TextBox, Label, ListBox, ComboBox, MenuStrip, Panel, GroupBox, CheckBox, RadioButton), их основные свойства, методы и события, примеры использования.

4.3. Лабораторные занятия

Номер раздела, темы	Наименование раздела, темы	Наименование лабораторной работы	Норматив времени, час.
			Очная форма обучения
1, 2	Переменные. Типы данных. Операторы. Условные и циклические конструкции	Лабораторная работа № 1. Действия с числами. Условные и циклические конструкции	1
3	Строки	Лабораторная работа № 2. Преобразование символьных величин.	1
4, 7	Структуры данных (списки, кортежи, словари, множества, диапазоны)	Лабораторная работа № 3. Структуры данных. Списки.	2
		Лабораторная работа № 4. Структуры данных. Множества	1
		Лабораторная работа № 7. Структуры данных. Словари	2

	Рубежный контроль 1.		2
5	Функции	Лабораторная работа № 5. Использование функций.	1
6	Регулярные выражения	Лабораторная работа № 6. Регулярные выражения	1
8	Работа с файлами и каталогами	Лабораторная работа № 8. Файлы	1
9	Работа с изображениями	Лабораторная работа № 9. Графика. Библиотека Pillow	2
	Рубежный контроль 2.		2
Всего			16
10, 11	Общие сведения о программах, лексемах и алфавите. Основные управляющие конструкции C++	Лабораторная работа №10. Действия с числами	6
12,13	Указатели и массивы в C++. Функции в C++.	Лабораторная работа №11. Работа с массивами	4
14	Строки и векторы в C++.	Лабораторная работа №12. Обработка строковой информации	6
15	Структуры и объединения в C++.	Лабораторная работа №13. Графика	2
		Лабораторная работа №14. Структуры	4
		Лабораторная работа №15. Объединения	2
16	Работа с файлами в C++	Лабораторная работа №16. Работа с файлами	8
Всего			32
Итого			48

4.4. Практические занятия

Номер раздела, темы	Наименование раздела, темы	Наименование практической работы	Норматив времени, час.
			Очная форма обучения
12	Указатели и массивы в C++.	Практическое занятие № 3. Особенности работы с указателями в C++.	4
13	Функции в C++.	Практическое занятие № 4. Особенности использования функций	4
	Рубежный контроль 3.		2
17	Обработка исключительных ситуаций в C++.	Практическое занятие № 5. Использование средств обработки исключительных ситуаций	4
	Рубежный контроль 4.		2
Всего			16
Итого			16

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

Лекционный курс базируется на пассивном методе обучения, реализующем традиционную объяснительно-иллюстративную образовательную технологию, в рамках которой обучающиеся выступают в роли слушателей, воспринимающих учебный материал, и участвующих в дискуссиях и экспресс-опросах.

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 и практических работ.

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

Залогом качественного выполнения лабораторных и практических работ является самостоятельная подготовка путем повторения материалов лекций. Рекомендуется подготовить вопросы по неясным моментам и обсудить их с преподавателем в начале каждого занятия.

Преподавателем запланировано применение на лабораторных и практических занятиях технологий развивающейся кооперации, коллективного взаимодействия, разбора конкретных ситуаций.

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 и практических 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 и практическим занятиям, к рубежным контролям (для очной формы обучения), подготовку к зачету.

Рекомендуемая трудоемкость самостоятельной работы представлена в таблице.

Наименование вида самостоятельной работы	Рекомендуемая трудоемкость, акад. час.
	Очная форма обучения
Самостоятельное изучение тем дисциплины:	30
Tkinter — это стандартная библиотека Python для создания графических интерфейсов (GUI).	10
Основы асинхронного программирования (asyncio). Разберитесь, что такое асинхронность, чем она отличается от многопоточности, как работают async/await, и попробуйте написать простой асинхронный клиент, который параллельно запрашивает данные с нескольких URL. Тема особенно полезна, если планируете делать высоконагруженные сервисы или парсеры.	10
Работа с базами данных и ORM (на примере SQLAlchemy). Освойте подключение к SQLite/PostgreSQL, создание таблиц, выполнение запросов, транзакции и миграции. Затем перейдите к ORM: научитесь описывать модели, делать выборки, связи «один-ко-многим» и т. д. В качестве проекта можно создать небольшое приложение (например, список задач или каталог товаров) с сохранением данных в БД.	10
Подготовка к лабораторным занятиям (по 1 часу на каждое занятие)	8
Подготовка к рубежным контролям (по 2 часа на каждый рубеж)	4
Подготовка к зачету	18
Всего:	60

Самостоятельное изучение тем дисциплины:	-
Подготовка к лабораторным занятиям (по 0,5 часа на каждое занятие (очная форма обучения))	8
Подготовка к рубежным контролям (по 1 часу на каждый рубеж)	2
Подготовка к практическим занятиям (по 1 часу на каждое занятие (очная форма обучения))	8
Подготовка к зачету	18
Всего:	36

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 (для очной формы обучения).
2. Отчеты обучающихся по лабораторным и практическим занятиям.
3. Банк тестовых заданий к рубежным контролям № 1 - № 4 (для очной формы обучения).
4. Список вопросов к зачету.

6.2. Система балльно-рейтинговой оценки работы обучающихся по дисциплине 3 семестр

№	Наименование	Содержание					
1	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы (доводятся до сведения обучающихся на первом учебном занятии)	Распределение баллов для зачета					
		Посещение лекций	Посещение лабораторных занятий	Выполнение лабораторных работ	Рубежный контроль № 1	Рубежный контроль № 2	Зачет
		До 16 баллов (16 * 1 балл = 16 б.)	До 8 баллов (8 * 1 балл = 8 б.)	До 28 баллов (3 * 5 балла + 1 * 6 баллов + 5 * 3 балла = 36 б.)	До 5 баллов	До 5 баллов	До 30 баллов
2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и зачета	- 60 и менее баллов – не зачтено (компетенции не освоены); - 61...73 – зачтено (пороговый уровень освоения компетенций); - 74...90 – зачтено (базовый уровень освоения компетенций); - 91...100 – зачтено (продвинутый уровень освоения компетенций).					

№	Наименование	Содержание
3	Критерии допуска к промежуточной аттестации, возможности получения автоматического зачета (экзаменационной оценки) по дисциплине, возможность получения бонусных баллов	Для допуска к промежуточной аттестации по дисциплине за семестр 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, 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 обучающегося могут быть начислены дополнительные баллы. Максимальное количество дополнительных баллов за академическую активность 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.
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 зачету 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 при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем

4 семестр

№	Наименование	Содержание						
1	Распределение баллов за семестры по видам учебной работы, сроки сдачи учебной работы (доводятся до сведения обучающихся на первом учебном занятии)	Распределение баллов для экзамена						
		Посещение лекций	Посещение лабораторных занятий	Выполнение лабораторных работ	Посещение практических занятий	Выполнение практических работ	Рубежный контроль № 3	Рубежный контроль № 4
		До 12 баллов (12 * 1 балл = 12 б.)	До 8 баллов (16 * 0,5 баллов = 8 б.)	До 24 баллов (2 * 4 балла, 1 * 6 баллов, 2 * 3 балла, 2 * 2 балла = 24 б.)	До 4 баллов (8 * 0,5 балл = 4 б.)	До 12 баллов (3 * 4 балла = 12 б.)	До 5 баллов	До 5 баллов
2	Критерий пересчета баллов в традиционную оценку по итогам работы в семестре и зачета	- 60 и менее баллов – не зачтено (компетенции не освоены); - 61...73 – зачтено (пороговый уровень освоения компетенций); - 74...90 – зачтено (базовый уровень освоения компетенций); - 91...100 – зачтено (продвинутый уровень освоения компетенций).						
3	Критерии допуска к промежуточной аттестации, возможности получения автоматического зачета (экзаменационной оценки) по дисциплине, возможность получения бонусных баллов	Для допуска к промежуточной аттестации по дисциплине за семестр 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, 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 обучающегося могут быть начислены дополнительные баллы. Максимальное количество дополнительных баллов за академическую активность 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.						

№	Наименование	Содержание
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 зачету 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 при переводе или восстановлении, проводится путем выполнения дополнительных заданий, форма и объем которых определяется преподавателем

6.3. Процедура оценивания результатов освоения дисциплины

Рубежные контроли проводятся в форме письменных ответов на вопросы. Каждому обучающемуся предлагается по 2 вопроса, за ответ на который обучающийся может получить до 2,5 баллов.

Перед проведением каждого рубежного контроля преподаватель прорабатывает основной материал соответствующих разделов дисциплины в форме краткой лекции-дискуссии. Примерные варианты тестовых заданий для 1 - 4 рубежного контроля приведены ниже. На каждое тестирование при рубежном контроле отводится 2 академических часа.

Преподаватель оценивает в баллах результаты тестирования по количеству правильных ответов и заносит в ведомость учета текущей успеваемости.

Зачет состоит из 2 вопросов и практического задания. Вопросы к зачету доводятся до обучающихся на последней лекции в семестре. Ответы на каждый вопрос оцениваются до 10 баллов, выполнение практического задания оценивается до 10 баллов. На подготовку ответа отводится 1 астрономический час.

Результаты текущего контроля успеваемости и зачета заносятся преподавателем в зачетно-экзаменационные ведомости, которые сдаются в организационный отдел института в день зачета, а также выставляются в зачетную книжку обучающегося.

6.4. Примеры оценочных средств для рубежных контролей и зачета

Примерные вопросы для рубежных контролей

Рубежный контроль 1: Список вопросов

1. Назовите основные типы данных
2. Из чего состоит любое выражение? К чему сводится любое выражение?
3. В чем различие между выражением и инструкцией?
4. Перечислите операторы сравнения.
5. В чем суть различия между оператором равенства и оператором присваивания?
6. Объясните, что такое условие и где используются условия.
7. Какую комбинацию клавиш нужно нажать, чтобы вывести программу из бесконечного цикла?
8. Чем различаются инструкции break и continue?
9. Чем различаются вызовы функций range (10), range (0,10) и range (0, 10, 1) в цикле for?
10. Напишите короткую программу, выводящую числа от 1 до 10 с помощью цикла for. Затем напишите аналогичную программу, в которой используется цикл while.
11. Что означают эти скобки: []?
12. Как бы вы присвоили значение 'hello' в качестве третьего элемента списка, хранящегося в переменной spam? (Предполагается, что в переменной spam содержится список [2, 4, 6, 8, 10].)
13. Переменная spam содержит список ['a', 'b', 'c', 'd']. Каково значение выражения spam[int ('3' * 2 / 11)] ?

14. Переменная `spam` содержит список `['a', 'b', 'c', 'd']`. Каково значение выражения `spam[-1]`?
15. Переменная `spam` содержит список `['a', 'b', 'c', 'd']`. Каково значение выражения `spam[:2]`?
16. Переменная `bacon` содержит список `[3.14, 'cat', 11, 'cat', True]`. Каково значение выражения `bacon.index('cat')`?
17. Переменная `bacon` содержит список `[3.14, 'cat', 11, 'cat', True]`. Как будет выглядеть список, хранящийся в переменной `bacon`, после следующего вызова: `bacon.append(98)` ?
18. Переменная `bacon` содержит список `[3.14, 'cat', 11, 'cat', True]`. Как будет выглядеть список, хранящийся в переменной `bacon`, после следующего вызова: `bacon.remove('cat')` ?
19. Какие операторы используются для конкатенации списков?
20. В чем состоит различие между предусмотренными для списков методами `append()` и `insert()`?
21. Назовите несколько общих признаков списков и строк.
22. Чем кортежи отличаются от списков?
23. Как бы вы записали кортеж, содержащий единственное значение в виде целого числа 42?
24. Как преобразовать список в кортеж? Как преобразовать кортеж в список?
25. Переменные, которые «содержат» список, на самом деле не содержат непосредственно сам список. Что же тогда они содержат?
26. Что такое экранированные символы?
27. Что представляют собой экранированные символы `\n` и `\t`?
28. Как добавить символ обратной косой черты (`\`) в строку?
29. Строковое значение `"How's Moving Castle"` — это допустимая строка. Почему она не вызовет ошибку, несмотря на наличие неэкранированного символа апострофа в слове `How's`?
30. Если вы не хотите вставлять символ `\n` и свою строку; то как вы напишете строку, содержащую символы новой строки?
31. Каковы будут результаты вычисления приведенных ниже выражений?
 - `'Hello world!'` `[1]`
 - `'Hello world!'` `[0:5]`
 - `'Hello world!'` `[:5]`
 - `'Hello world!'` `[3:]`
32. Каковы будут результаты вычисления приведенных ниже выражений?
 - `'Hello'.upper()`
 - `'Hello'.upper().isupper()`
 - `'Hello'.upper().lower()`
33. Каковы будут результаты вычисления приведенных ниже выражений?
 - `'Remember, remember, the fifth of November.'.split()`
 - `'-'.join('There can be only one.'.split())`
34. Как удалить пробельные символы в начале и конце строки?

Рубежный контроль 2: Список вопросов

1. Что дает использование функций в программах?
2. Когда именно выполняется код функции: когда она определяется или когда вызывается?
3. С помощью какой инструкции создаются функции?
4. Чем отличается определение функции от ее вызова?

5. Сколько глобальных областей видимости может иметь программа на языке Python? Сколько локальных?
6. Что происходит с переменными, находящимися в локальной области видимости, при возврате из функции?
7. Что такое возвращаемое значение? Может ли возвращаемое значение быть частью выражения?
8. Каково возвращаемое значение функции, если в ней отсутствует инструкция `return`?
9. Как заставить переменную в функции ссылаться на глобальную переменную?
10. Что такое тип данных `None`?
11. Какой код помещается в блок `try`? Какой код помещается в блок `except`?
12. Относительно чего задается относительный путь?
13. С чего начинается абсолютный путь?
14. Каково назначение функций `os.getcwd()` и `os.chdir()`?
15. Что собой представляют папки `.` и `..`?
16. Назовите три возможных значения аргумента, задающие режим открытия файла, которые могут передаваться функции `open()`.
17. Что происходит при открытии существующего файла в режиме записи?
18. Чем различаются методы `read()` и `readlines()`?
19. Какая функция используется для переименования файлов?
20. Что такое класс? Что такое объект (экземпляр класса)? Что такое поле класса?
21. Общая структура описания класса. Назначение аргумента `self`.
22. Как создается экземпляр класса?
23. Что такое конструктор класса?
24. Что такое атрибуты объекта класса? Когда их можно добавлять, изменять?
25. Как осуществляется вызов метода класса?
26. Каково назначение наследования? Как соотносятся между собой базовый и производный классы?
27. Можно ли изменять унаследованные методы?
28. Как загрузить готовое изображение?
29. Как создать новое изображение?
30. Перечислите основные методы, предназначенные для работы с изображением.
31. Класс `Draw`, его назначение. Основные методы этого класса.
32. Вывод текста на графическое изображение.
33. Как выглядит код для пустого словаря?
34. Как выглядит элемент словаря с ключом `"foo"` и значением `42`?
35. Опишите основные различия между словарем и списком.
36. Что произойдет при попытке получения доступа к элементу `spam["foo"]`, если `spam` — это `{ "bar": 100 }`?
37. Если в переменной `spam` хранится словарь, то в чем состоит разница между выражениями `'cat' in spam` и `'cat' in spam.keys()`?
38. Если в переменной `spam` хранится словарь, то в чем состоит разница между выражениями `'cat' in spam` и `'cat' in spam.values()`?
39. Что возвращает метод `search()`?
40. В синтаксисе регулярных выражений круглые скобки и точки имеют особый смысл. Как бы вы указали в регулярном выражении, что символы круглых скобок и точки сами являются объектом поиска?
41. Что означает символ `|` в регулярных выражениях?
42. Какие две функции выполняет символ `?` в регулярных выражениях?
43. В чем разница между символами `+` и `*` в регулярных выражениях?
44. В чем разница между записями `{3}` и `{3,5}` в регулярных выражениях?

45. Что означают сокращенные символьные классы `\d`, `\w` и `\s` в регулярных выражениях?
46. Что означают сокращенные символьные классы `\D`, `\W` и `\S` в регулярных выражениях?
47. Как сделать регулярные выражения нечувствительными к регистру?

Рубежный контроль 3: Список вопросов

1. Что напечатает следующая программа?

```
#include<iostream.h>
main ()
{
    int x;
    x = -3+4*5-6; cout<<x<<" "; /* Операция 1 */
    x = 3+4%5-6; cout<<x<<" "; /* Операция 2 */
    x = -3*4%-6/5; cout<<x<<" "; /* Операция 3 */
    x = (7+6)%5/2; cout<<x<<" "; /* Операция 4 */
}
```

2. Что напечатает следующая программа.

```
#include<iostream.h>
#include<stdio.h>
main()
{
    int a,b,c,v,k;
    cout<<"Задайте целое число: ";
    cin>>v;
    k=v;
    cout<<"      До   Во время   После"<<endl;
    v=k;a=v;b=v++;c=v;printf("v++%8d%8d%8d\n",a,b,c);
    v=k;a=v;b=v--;c=v;printf("v--%8d%8d%8d\n",a,b,c);
    v=k;a=v;b=++v;c=v;printf("++v%8d%8d%8d\n",a,b,c);
    v=k;a=v;b=--v;c=v;printf("--v%8d%8d%8d\n",a,b,c);
}
```

3. Проиллюстрируйте в программе применение логических операций и операций увеличения.
4. Выдать на печать в обратном порядке цифры целого положительного числа N.
5. Напишите программу для замены в слове X всех букв «а» на сочетание «ку».
6. Напишите программу, удваивающую каждую букву слова X.
7. Вычеркните из слова X буквы, стоящие на четных местах.
8. Составить массив, каждый элемент которого равен максимальному из соответствующих значений двух других массивов.

Рубежный контроль 4: Список вопросов

1. Что располагается между служебным словом `struct` и открывающей фигурной скобкой «{»? В каком случае эта конструкция может отсутствовать?
2. Каков в общем виде шаблон структуры?
3. Способы задания экземпляра структуры.
4. Способы обращения к элементам структур.
5. В чем заключается разница между ссылками и указателями?
6. Перечислить функции, осуществляющие чтение из файла.
7. Перечислить функции, осуществляющие запись в файл.
8. В чем заключается отличие форматного ввода/вывода от обычного?
9. Организация стандартного ввода/вывода

10. Назначение класса `fstream`. Его основные методы.
11. Назначение класса `ofstream`. Его основные методы.
12. Назначение класса `ifstream`. Его основные методы.
13. Понятие бинарного файла. Примеры работы с ним.

Примерный список вопросов к зачету.

1 семестр

1. Основные типы переменных в Python. Именованые переменных. Преобразование типов. Удаление переменной.
2. Операторы. Двоичные операторы. Операторы работы с последовательностями. Приоритет выполнения операторов.
3. Условные конструкции.
4. Циклические конструкции.
5. Операторы `break` и `continue`.
6. Строки. Создание строки. Основные операции над строками.
7. Поиск и замена в строке.
8. Регулярные выражения. Синтаксис регулярных выражений.
9. Регулярные выражения. Поиск первого совпадения с шаблоном.
10. Регулярные выражения. Поиск всех совпадений с шаблоном.
11. Регулярные выражения. Замена в строке.
12. Списки. Различные способы задания списков.
13. Основные операции над списками.
14. Добавление и удаление элементов списка.
15. Поиск элемента в списке. Переворачивание и сортировка списка. Преобразование списка в строку.
16. Множества. Создание множества. Основные операции над множествами.
17. Словари. Создание словаря. Основные операции над словарями.
18. Кортежи. Создание кортежа. Основные операции с кортежами.
19. Функции, определяемые пользователем. Особенности определения функций. Локальные и глобальные параметры.
20. Анонимные функции. Рекурсия.
21. Декораторы функций. Анонимные функции.
22. Модули, их назначение. Стандартные модули, их краткая характеристика.
23. Понятие класса и объекта. Создание класса и объекта в Python.
24. Наследование. Реализация наследования.
25. Перегрузка операторов.
26. Обработка исключений. Инструкция `try... except`. Классы встроенных исключений.
27. Понятие файла. Основные методы работы с файлами.
28. Основные методы работы с каталогами. Права доступа к файлам и каталогам.
29. Библиотека `Pillow`. Создание нового изображения и загрузка готового изображения.
30. Библиотека `Pillow`. Рисование линий и фигур.

2 семестр

1. Алфавит языка C++. Лексемы языка C++.
2. Знаки операций в языке C++.

3. Скалярные типы и выражения в языке C++. Примеры использования простейших типов в языке C++.

4. Арифметические операции в языке C++. Примеры.

5. Инкремент и декремент. Примеры.

6. Операции присваивания и отношения. Примеры.

7. Логические операции. Примеры.

8. Условная операция. Примеры.

9. Операция sizeof. Примеры.

10. Операции преобразования типов. Примеры.

11. Оператор if...else. Примеры.

12. Оператор switch. Примеры.

13. Цикл while. Примеры.

14. Цикл do...while. Примеры.

15. Цикл for. Примеры.

16. Оператор безусловного перехода. Примеры.

17. Оператор принудительного выхода из цикла или переключателя. Примеры.

18. Оператор завершения выполнения текущего шага тела цикла. Примеры.

Примерный список заданий к зачету.

Пример 1. Действия с числами.

1. Найти несколько простых чисел Фибоначчи.
2. Для заданного целого числа m найти среди первых $m \cdot m - 1$ чисел Фибоначчи хотя бы одно, делящееся на m .
3. Вычислить $(N)!!$, где $(2N)!! = 2 \cdot 4 \cdot 6 \cdot \dots \cdot 2N$, $(2N+1)!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot (2N+1)$.

Пример 2. Работа с массивами

1. Из заданного множества точек на плоскости выбрать две различные точки так, чтобы количества точек, лежащих по разные стороны прямой, проходящей через эти две точки, различались наименьшим образом.
2. Определить радиус и центр окружности, на которой лежит наибольшее число точек заданного на плоскости множества точек.
3. Определить радиус и центр окружности минимального радиуса, проходящей хотя бы через три различные точки заданного множества точек на плоскости.

Пример 3. Обработка символьной информации.

1. Перечислить все слова заданного предложения, которые состоят из тех же букв, что и первое слово предложения.
2. В заданном предложении найти пару слов, из которых одно является обращением другого.
3. Для каждого из слов заданного предложения указать, сколько раз оно встречается в предложении.

Пример 4. Файлы.

1. Дан файл f , компоненты которого являются целыми числами. Найти: 1) сумму компонент файла, 2) произведение компонент файла, 3) сумму квадратов компонент файла, 4) модуль суммы и квадрат произведения компонент файла, 5) последнюю компоненту файла.
2. Дан символьный файл f . Получить копию файла в файле g .
3. Дан символьный файл f . Получить в файле g компоненты файла f в обратном порядке.

Пример 5. Структуры.

1. Опишите, используя структуру, телефонную книгу.

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 учебная литература

1 Андреева, О. В. Основы алгоритмизации и программирования на языке Python : учебник / О. В. Андреева, О. И. Ремизова. - Москва : Издательский Дом НИТУ «МИСиС», 2022. - 149 с. - ISBN 978-5-907560-22-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/>. – Режим доступа: по подписке.

2. Златопольский, Д.М. Основы программирования на языке Python / Д.М. Златопольский. - Москва : ДМК Пресс, 2021. - 284 с. - ISBN 978-5-97060-552-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/102814>. – Режим доступа: по подписке.

3 Медведев А.А. Изучение языка программирования Python [Электронный ресурс]: методические рекомендации для подготовки бакалавров и специалистов направлений 09.03.03, 09.03.04, 10.05.01, 10.05.03 «Прикладная информатика», «Программная инженерия», «Компьютерная безопасность», «Информационная безопасность» / Министерство науки и высшего образования Российской Федерации, Курганский государственный университет, Кафедра программного обеспечения автоматизированных систем; [сост.: А.А. Медведев]. - Электрон. текстовые дан. (тип файла: pdf; размер: 488 Kb). - Курган: Издательство Курганского государственного университета, 2019. - 43, [1] с.: рис. - Библиогр.: с. 43.

4. Гуриков, С. Р. Основы алгоритмизации и программирования на Visual C++ : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2022. — 515 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/1039154. - ISBN 978-5-16-015500-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1039154>. – Режим доступа: по подписке.

5 Гуриков, С. Р. Основы алгоритмизации и программирования на Visual C++ : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2022. — 515 с. — (Высшее образование: Бакалавриат). — DOI 10.12737/1039154. - ISBN 978-5-16-015500-5. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1039154>. – Режим доступа: по подписке.

7.2 Дополнительная литература

1. Жуков, Р. А. Язык программирования Python: практикум : учебное пособие / Р.А. Жуков. — Москва : ИНФРА-М, 2021. — 216 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Бакалавриат). — DOI 10.12737/textbook_5cb5ca35aaa7f5.89424805. - ISBN 978-5-16-016971-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1412168>. – Режим доступа: по подписке.

2. Русанова, Я. М. C++ как второй язык в обучении приемам и технологиям программирования: учеб. пособие / Я. М. Русанова. - Ростов-на-Дону: Издательство ЮФУ, 2010. - 200 с. - ISBN 978-5-9275-0749-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/550811>. – Режим доступа: по подписке.

3. Гридчин, А. В. Информационные технологии. Программирование на C++ : учебно-методическое пособие / А. В. Гридчин. - Новосибирск : Изд-во НГТУ, 2020. - 68 с. - ISBN 978-5-7782-4174-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1866900>. – Режим доступа: по подписке.

4. Ашарина, И. В. Язык C++ и объектно-ориентированное программирование в C++. Лабораторный практикум: Учебное пособие для вузов / Ашарина И.В., Крупская Ж.Ф. - Москва :Гор. линия-Телеком, 2016. - 232 с. - ISBN . - Текст : электронный. - URL: <https://new.znanium.com/catalog/product/973780>

8. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ ОБУЧАЮЩИХСЯ

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. Кувшинов, Д. Р. Компьютерные науки : Основы программирования: Учебное пособие / Кувшинов Д.Р., - 2-е изд., стер. - Москва :Флинта, Изд-во Урал. ун-та, 2017. - 102 с. ISBN 978-5-9765-3144-4. - Текст : электронный. - URL: <https://new.znanium.com/catalog/product/948144>
3. Культин, Н. Б. C/C++ в задачах и примерах / Н. Б. Культин. — Санкт-Петербург : БХВ-Петербург, 2015. — 285 с. - ISBN 978-5-9775-3322-5. - Текст : электронный. - URL: <https://new.znanium.com/catalog/product/940363>

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

1. Сайт дистанционного обучения в НОУ (Национальный Открытый Университет) «ИНТУИТ» содержит бесплатные курсы, программы повышения квалификации и профессиональной переподготовки, интересные доклады и другую полезную информацию <http://www.intuit.ru>.
2. Федеральный портал «Российское образование» <http://www.edu.ru/>
3. Информационный сайт, содержащий справочные материалы по информатике, которые включают в себя курс лекций, схемы, презентации, рефераты и др. informatikaplus.narod.ru.
4. Постоянно обновляемый электронный учебник (свободный доступ), содержащий полную информацию о языке программирования Python. <https://docs.python.org/3/tutorial/index.html>
5. Сайт, содержащий необходимые дистрибутивы и полную информацию для языка программирования Python <https://www.python.org/>
6. Сайт кафедры ПОАС КГУ «Информатика и программирование: шаг за шагом» <http://it.kgsu.ru/>.

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 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 обучающихся.

Аннотация к рабочей программе дисциплины
«ОСНОВЫ ПРОГРАММИРОВАНИЯ»

образовательной программы высшего образования –
программы бакалавриата

35.03.06 - Агроинженерия

Направленность:

Информационные технологии в электроэнергетике

Трудоемкость дисциплины: 6 з.е. (216 академических часа)

Семестр: 3, 4 (очная),

Форма промежуточной аттестации: зачет, зачет

Содержание дисциплины

Краткая характеристика языка программирования Python. Условные и циклические конструкции. Работа со строками. Списки, множества, кортежи, диапазоны. Организация функций. Модули. Файлы. Графика.

Краткая характеристика языка программирования C/C++. Условные и циклические конструкции. Массивы. Работа со строками. Функции. Файлы. Создание консольных приложений и приложений Windows Forms. CLR-приложения. Регулируемые или нерегулируемые указатели. Основные компоненты, используемые при создании приложений Windows Forms.

ЛИСТ
регистрации изменений (дополнений) в рабочую программу
учебной дисциплины
«ОСНОВЫ ПРОГРАММИРОВАНИЯ»

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

Ответственный преподаватель _____ / _____ /

Изменения утверждены на заседании кафедры «__» _____ 20__ г.,
Протокол № ____

Заведующий кафедрой _____ «__» _____ 20__ г.

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

Ответственный преподаватель _____ / _____ /

Изменения утверждены на заседании кафедры «__» _____ 20__ г.,
Протокол № ____

Заведующий кафедрой _____ «__» _____ 20__ г.