

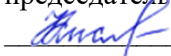
Документ подписан электронной подписью  
 Информация о владельце:  
 ФИО: Кислова Наталья Николаевна  
 Должность: Проректор по УМР и качеству образования  
 Дата подписания: 19.03.2024  
 Уникальный программный ключ:  
 52802513f5b14a975b7e9b13008093d5726b159bf6064f865ae65b96a966c035

**МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение  
 высшего образования  
 «Самарский государственный социально-педагогический университет»**

**Кафедра информатики, прикладной математики и методики их преподавания**

УТВЕРЖДАЮ

Проректор по УМР и КО,  
 председатель УМС СГСПУ  
  
 Н.Н. Кислова

**МОДУЛЬ "ПРЕДМЕТНОЕ ОБУЧЕНИЕ.  
 ИНФОРМАТИКА"  
 Программирование  
 рабочая программа дисциплины (модуля)**

Закреплена за кафедрой	<b>Информатики, прикладной математики и методики их преподавания</b>		
Учебный план	ФМФИ-620ИДо(5г) Направление подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки) Направленность (профиль): «Информатика» и «Дополнительное образование (в области информатики и ИКТ)»		
Квалификация	<b>бакалавр</b>		
Форма обучения	<b>очная</b>		
Общая трудоемкость	<b>32 ЗЕТ</b>		
Часов по учебному плану	1152	Виды контроля в семестрах:	
в том числе:		экзамены 4, 5, 6, 7, 8	
аудиторные занятия	467	зачеты с оценкой 3	
самостоятельная работа	685	курсовые проекты 6	

**Распределение часов дисциплины по семестрам**

Семестр(Курс.Номер семестра на курсе)	3(2.1)		4(2.2)		5(3.1)		6(3.2)		7(4.1)		8(4.2)		Итого	
	УП	РПД	УП	РПД	УП	РПД	УП	РПД	УП	РПД	УП	РПД	УП	УП
Лекции	22	22	22	22	22	22	22	22	46	46	46	46	180	180
Консультации	0	0	0	0	0	0	3	3	0	0	0	0	3	3
Лабораторные	36	36	36	36	36	36	36	36	70	70	70	70	284	284
В том числе инт.	14	14	14	14	14	14	14	14	30	30	32	32	104	104
Итого ауд.	58	58	58	58	58	58	61	61	116	116	116	116	467	467
Контактная работа	58	58	58	58	58	58	61	61	116	116	116	116	467	467
Сам. работа	86	86	86	86	86	86	83	83	158	158	158	158	685	685
Итого	144	144	144	144	144	144	144	144	288	288	288	288	1152	1152

Программу составил(и):

Тюжина Ирина Викторовна

При наличии обучающихся из числа лиц с ограниченными возможностями здоровья, которым необходим особый порядок освоения дисциплины (модуля), по их желанию разрабатывается адаптированная к ограничениям их здоровья рабочая программа дисциплины (модуля).

Рабочая программа дисциплины

### **Программирование**

разработана в соответствии с ФГОС ВО:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки) (приказ Минобрнауки России от 22.02.2018 г. № 125)

составлена на основании учебного плана:

Направление подготовки: 44.03.05 Педагогическое образование (с двумя профилями подготовки)

Направленность (профиль) «Информатика» и «Дополнительное образование (в области информатики и ИКТ)»

утвержденного Учёным советом СГСПУ от 30.08.2019 протокол № 1.

Рабочая программа одобрена на заседании кафедры

### **Информатики, прикладной математики и методики их преподавания**

Протокол от 27.08.2019 г. №1

Переутверждена на основании решения Ученого совета СГСПУ

Протокол заседания Ученого совета СГСПУ от 25.02.2022 г. №7.

Зав. кафедрой Т.В. Добудько

Начальник УОП

Н.А. Доманина

### 1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

**Цель изучения дисциплины:** формирование у обучающихся систематизированных знаний и навыков в области программирования, ознакомление с различными парадигмами программирования.

**Задачи изучения дисциплины:** ознакомить студентов с основными конструкциями различных языков программирования, основными типами и структурами данных; формирование умений разработки, отладки и тестирования программ.

**Область профессиональной деятельности:** 01 Образование и наука

### 2. МЕСТО ДИСЦИПЛИНЫ (МОДУЛЯ) В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

Цикл (раздел) ОП: Б1.О.07

#### 2.1 Требования к предварительной подготовке обучающегося:

Содержание дисциплины базируется на материале:

Информационные технологии и системы

Программное обеспечение электронно-вычислительной машины

#### 2.2 Дисциплины (модули) и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:

Программирование в образовательной робототехнике, Основы искусственного интеллекта

Интернет-программирование

Разработка приложений для мобильных устройств

Олимпиадные задачи по информатике

Производственная практика (педагогическая практика) (по профилю "Информатика")

### 3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

**УК-1 Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий**

**УК-1.1. Выявляет проблемную ситуацию в процессе анализа проблемы, определяет этапы ее разрешения с учетом вариативных контекстов**

Знает: этапы решения задачи на компьютере.

Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи.

**УК-1.2. Находит, критически анализирует и выбирает информацию, необходимую для выработки стратегии действий по разрешению проблемной ситуации**

Знает: способы формализации алгоритмов на языках программирования высокого уровня.

Умеет: осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи.

**УК-1.3. Рассматривает различные варианты решения задачи, оценивает их преимущества и риски**

Знает: методы процедурного, объектно-ориентированного, функционального и визуального программирования; основные конструкции языков программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени.

Умеет: строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня.

**УК-1.4. Грамотно, логично, аргументированно формулирует собственные суждения и оценки, предлагает стратегию действий**

Знает: основные виды ошибок, возникающих при решении задачи.

Умеет: комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.

**УК-1.5. Определяет и оценивает практические последствия возможных вариантов решения задачи**

Умеет: выполнять оценку сложности алгоритмов, проводить анализ и оценивание полученных результатов.

### 4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр	Часов	Интеракт.
	<b>Раздел 1. Процедурное программирование</b>			
1.1	Общая характеристика и основные понятия процедурного программирования, и его изучение в школьном курсе информатики /Лек/	3	2	0
1.2	Типы данных /Лек/	3	2	0
1.3	Разработка и оформление программы /Лек/	3	4	0
1.4	Основные операторы языка программирования /Лек/	3	10	0
1.5	Структурирование программ. Процедуры и функции /Лек/	3	4	0
1.6	Вычисления по формулам /Лаб/	3	36	14
1.7	Вычисления по формулам /Ср/	3	6	0
1.8	Условный оператор /Ср/	3	6	0
1.9	Логические операции /Ср/	3	6	0
1.10	Оператор цикла с параметром /Ср/	3	6	0
1.11	Элементарные задачи целочисленной арифметики /Ср/	3	10	0

1.12	Оператор цикла с предусловием /Ср/	3	6	0
1.13	Оператор цикла с постусловием /Ср/	3	6	0
1.14	Организация циклов (различных) /Ср/	3	6	0
1.15	Перечислимый и ограниченный типы данных. /Ср/	3	6	0
1.16	Оператор выбора /Ср/	3	6	0
1.17	Структурирование программ с использованием процедур /Ср/	3	6	0
1.18	Структурирование программ с использованием функций /Ср/	3	6	0
1.19	Рекуррентные соотношения /Ср/	3	10	0
<b>Раздел 2. Структуры данных</b>				
2.1	Обработка строковых величин /Лек/	4	4	0
2.2	Обработка массивов /Лек/	4	4	0
2.3	Множественный тип данных /Лек/	4	2	0
2.4	Записи /Лек/	4	4	0
2.5	Файлы /Лек/	4	4	0
2.6	Графика /Лек/	4	4	0
2.7	Структуры данных: строки /Лаб/	4	36	14
2.8	Строковый тип данных /Ср/	4	20	0
2.9	Массивы /Ср/	4	20	0
2.10	Множества /Ср/	4	10	0
2.11	Записи /Ср/	4	10	0
2.12	Файлы /Ср/	4	10	0
2.13	Графика /Ср/	4	16	0
<b>Раздел 3. Мультипарадигмальное программирование</b>				
3.1	Основные возможности мультипарадигмальных языков программирования, структура программы/Лек/	5	2	0
3.2	Форматы представления чисел /Лек/	5	4	0
3.3	Структуры данных: списки, множества, кортежи, строки, словари /Лек/	5	6	0
3.4	Основные операторы /Лек/	5	4	0
3.5	Работа с файлами /Лек/	5	2	0
3.6	Функции и обработка исключений /Лек/	5	4	0
3.7	Вычисления по формулам /Лаб/	5	36	14
3.8	Вычисления по формулам /Ср/	5	10	0
3.9	Условный оператор «IF ... ELIF ELSE» /Ср/	5	6	0
3.10	Циклические алгоритмы, инструкция «FOR» /Ср/	5	6	0
3.11	Элементарные задачи целочисленной арифметики /Ср/	5	6	0
3.12	Процедуры обработки списков /Ср/	5	6	0
3.13	Процедуры обработки слов /Ср/	5	6	0
3.14	Процедуры обработки множеств /Ср/	5	6	0
3.15	Кортежи /Ср/	5	6	0
3.16	Работа с файлами /Ср/	5	6	0
3.17	Структурирование программ с использованием функций /Ср/	5	14	0
3.18	Рекуррентные соотношения /Ср/	5	6	0
3.19	Словари /Ср/	5	8	0
<b>Раздел 4. Подключаемые библиотеки</b>				
4.1	Модули /Лек/	6	2	0
4.2	Библиотеки для работы с веб-данными /Лек/	6	4	0
4.3	Графический интерфейс: библиотека Tkinter /Лек/	6	4	0
4.4	Графический интерфейс: библиотека PyGame /Лек/	6	4	0
4.5	Фреймворк для создания веб-приложений: Flask /Лек/	6	4	0
4.6	Голосовой интерфейс /Лек/	6	4	0
4.7	Модули /Лаб/	6	36	14
4.8	Модули /Ср/	6	20	0
4.9	Библиотеки для работы с веб-данными /Ср/	6	10	0
4.10	Графический интерфейс: библиотека Tkinter /Ср/	6	10	0
4.11	Графический интерфейс: библиотека PyGame /Ср/	6	10	0
4.12	Фреймворк для создания веб-приложений: Flask /Ср/	6	20	0
4.13	Голосовой интерфейс /Ср/	6	13	0
4.14	Курсовой проект /Конс/	6	3	0
<b>Раздел 5. Объектно-ориентированное программирование</b>				
5.1	Элементарные структуры данных /Лек/	7	4	0
5.2	Абстрактные типы данных /Лек/	7	4	0

5.3	Рекурсия и деревья /Лек/	7	4	0
5.4	Методы сортировки /Лек/	7	4	0
5.5	Таблицы символов и деревья бинарного поиска /Лек/	7	4	0
5.6	Сбалансированные деревья /Лек/	7	4	0
5.7	Хэширование /Лек/	7	4	0
5.8	Поразрядный поиск /Лек/	7	6	0
5.9	Внешний поиск /Лек/	7	6	0
5.10	Поиск на графе /Лек/	7	6	0
5.11	Элементарные структуры данных /Лаб/	7	70	30
5.12	Элементарные структуры данных /Ср/	7	14	0
5.13	Абстрактные типы данных /Ср/	7	14	0
5.14	Рекурсия и деревья /Ср/	7	14	0
5.15	Элементарные методы сортировки /Ср/	7	14	0
5.16	Поразрядная сортировка /Ср/	7	14	0
5.17	Специальные методы сортировки /Ср/	7	14	0
5.18	Таблицы символов и деревья бинарного поиска /Ср/	7	14	0
5.19	Сбалансированные деревья /Ср/	7	14	0
5.20	Хэширование /Ср/	7	14	0
5.21	Поразрядный поиск /Ср/	7	14	0
5.22	Внешний поиск /Ср/	7	16	0
5.23	Поиск на графе /Ср/	7	16	0
<b>Раздел 6. Объектно-ориентированное программирование</b>				
6.1	Введение в объектно-ориентированное программирование /Лек/	8	4	0
6.2	Конструкции классов и области видимости /Лек/	8	4	0
6.3	Конструктор и деструктор /Лек/	8	4	0
6.4	Использование указателей в классе /Лек/	8	4	0
6.5	Наследование /Лек/	8	4	0
6.6	Полиморфизм /Лек/	8	4	0
6.7	Виды классов /Лек/	8	4	0
6.8	Исключительные ситуации и их обработка /Лек/	8	4	0
6.9	Перезагрузка операций в классах /Лек/	8	4	0
6.10	Стандартная библиотека шаблонов STL /Лек/	8	4	0
6.11	Потоковый ввод и вывод /Лек/	8	6	0
6.12	Основы ООП /Лаб/	8	70	32
6.13	Потоковый ввод и вывод /Ср/	8	14	0
6.14	Основы ООП /Ср/	8	14	0
6.15	Конструкции классов и области видимости /Ср/	8	14	0
6.16	Конструктор и деструктор /Ср/	8	14	0
6.17	Использование указателей в классе /Ср/	8	14	0
6.18	Наследование /Ср/	8	14	0
6.19	Полиморфизм /Ср/	8	14	0
6.20	Виды классов /Ср/	8	14	0
6.21	Параметризованные функции /Ср/	8	14	0
6.22	Исключительные ситуации и их обработка /Ср/	8	14	0
6.23	Перезагрузка операций в классах /Ср/	8	14	0
6.24	Стандартная библиотека шаблонов STL /Ср/	8	14	0
6.25	Потоковый ввод и вывод /Ср/	8	4	0
<b>5. Оценочные и методические материалы по дисциплине (модулю)</b>				
<b>5.1. Содержание аудиторной работы по дисциплине (модулю)</b>				
<b>3 семестр, 11 лекций, 18 лабораторных занятий</b>				
<b>Раздел 1. Процедурное программирование</b>				
Лекция №1 (2 часа)				
Общая характеристика и основные понятия процедурного программирования, и его изучение в школьном курсе информатики				
Вопросы и задания:				
Процедурный язык программирования. Назначение. Возможности. Стандарт языка и его реализации. Алфавит языка. Операторы. Имена и идентификаторы. Переменные. Арифметические операции. Встроенные функции. Функции и процедуры. Изучение программирования в школьном курсе информатики.				
Лекция №2 (2 часа)				
Типы данных				
Вопросы и задания:				

Скалярные типы данных: целочисленные, вещественные, литерные, булевские, перечисляемый, интервальный.  
Структурированные типы данных.

Лекция №3-4 (4 часа)  
Разработка и оформление программы

Вопросы и задания:

Структура программы. Разделы описания типов, меток, переменных. Раздел процедур и функций. Тело программы. Обязательные элементы программы. Понятие алгоритма. Блок схема алгоритма.

Лекции №5-9 (10 часов)  
Основные операторы языка программирования

Вопросы и задания:

Запись условного оператора: полная и сокращенная формы. Операторные скобки: Begin end. Применение условного оператора. Оператор цикла с параметром «for». Циклические конструкции. Структура оператора. Прямой и обратный цикл. Ограничения, накладываемые на параметр цикла. Применение оператора «for». Операторы цикла с условием. Оператор с предусловием «While». Оператор с постусловием «Repeat... Until». Перечислимый и ограниченный типы данных. Оператор выбора. Пользовательские типы данных.

Лекция №10-11 (4 часа)  
Структурирование программ. Процедуры и функции

Вопросы и задания:

Понятие подпрограммы. Параметры подпрограммы. Процедуры. Функции. Локальные переменные. Передача параметров. Рекурсия в программировании.

Лабораторные занятия №1-18 (36 часов)  
Вычисления по формулам

Вопросы и задания:

1. Вычисления по формулам
2. Условный оператор IF ... THEN
3. Логические операции...
4. Оператор цикла с параметром
5. Элементарные задачи целочисленной арифметики
6. Оператор цикла с предусловием
7. Оператор цикла с постусловием
8. Организация циклов (различных)
9. Перечислимый и ограниченный типы данных.
10. Оператор выбора
11. Структурирование программ с использованием процедур
12. Структурирование программ с использованием функций
13. Рекуррентные соотношения

**4 семестр, 11 лекций, 18 лабораторных занятий**

**Раздел 2. Структуры данных**

Лекция №12-13 (4 часа)  
Обработка строковых величин

Вопросы и задания:

Строковый тип данных. Процедуры и функции, определенные над строковым типом данных. Перевод строки в число и обратно. Применение строковых величин.

Лекции №14-15 (4 часа)  
Обработка массивов

Вопросы и задания:

Понятие массива. Ввод элементов массива: с клавиатуры и случайным образом. Обработка одномерных массивов. Сортировка. Описание двумерных массивов. Двумерные массивы: вывод на экран в виде матрицы. Решение типовых задач.

Лекция №16 (2 часа)  
Множественный тип данных

Вопросы и задания:

Понятие множества. Операции, определенные над множествами. Решение типовых задач.

Лекции №17-18 (14 часа)  
Записи

Вопросы и задания:

Тип данных «Record». Обращение к полю записи. Оператор «with». Обработка записей. Решение типовых задач.

Лекции №19-20 (4 часа)  
Файлы

Вопросы и задания:

Файловые переменные, оператор «assign». Типы файлов: текстовые, типизированные, нетипизированные. Обработка файлов. Чтение данных из файла, процедуры «reset», «read», «readln», «seek», «close». Запись данных в файл, процедуры «rewrite», «write», «writeln», «append».

Лекции №21-22 (4 часа)  
Графика

Вопросы и задания:

Обработка графики в процедурных языках программирования. Графические примитивы. Управление цветом. Построение графиков в различных системах координат. Анимация.

Лабораторные занятия №19-36 (36 часов)

Структуры данных: строки

Вопросы и задания:

1. Структуры данных: строки
2. Структуры данных: одномерные массивы
3. Структуры данных: двумерные массивы
4. Структура данных: множества.
5. Структура данных: файлы
6. Структура данных: записи.
7. Графические примитивы
8. График функций в декартовой системе координат
9. График функций в полярной системе координат
10. Построение кривых, заданных в параметрической форме.
11. Создание рисунков.
12. Имитация движения с помощью видеостраниц.
13. Избранные задачи.

**5 семестр, 11 лекций, 18 лабораторных занятий**  
**Раздел 3. Мультипарадигмальное программирование**

Лекция №23 (2 часа)

Основные возможности мультипарадигмальных языков программирования, структура программы

Вопросы и задания:

Понятие интерпретируемого языка программирования. Динамическая типизация. Синтаксис языка. Вложенные структуры.

Лекция №24-25 (4 часа)

Форматы представления чисел

Вопросы и задания:

Категории числовых типов. Перечень числовых типов. Представление целых чисел (литералов) в разных системах исчисления. Представление целых чисел в Python 3.0 Представление вещественных чисел. Комплексные числа. Создание комплексных чисел. Функция complex(). Значения NaN и None.

Лекция №26-28 (6 часов)

Структуры данных: списки, множества, кортежи, строки, словари

Вопросы и задания:

Списки: объявление и обработка. Индексы, отрицательное индексирование. Срезы. Кортежи: объявление и обработка. Операции над кортежами и примеры их применения. Строковый тип данных: объявление и обработка. Функции. Примеры применения. Регулярные выражения. Множества: объявление и обработка. Понятие «замороженного» множества. Словари: объявление и обработка. Пара «ключ-значение», способы обращения к значениям. Примеры применения.

Лекции №29-30 (4 часа)

Основные операторы

Вопросы и задания:

Арифметические операторы. Ассоциативность. Операторы сравнения (реляционные). Операторы присваивания. Побитовые операторы. Логические операторы. Операторы членства. Операторы тождественности

Лекция №31 (2 часа)

Работа с файлами

Вопросы и задания:

Файл и файловая переменная. Режим работы: чтение, запись, дозапись. Бинарные файлы. Базовые файловые методы. Стандартный ввод/вывод. Произвольный доступ. Построчная работа с файлами. Закрытие файла. Оператор with. Итерация.

Лекция №32-33 (4 часа)

Функции и обработка исключений

Вопросы и задания:

Именные и анонимные функции, инструкции «def», «return» и «lambda», функции с переменным числом аргументов, необязательные аргументы. Базовые типы исключений. Обработка исключений. Конструкция try/except. Запись try/except/finally. Оператор else совместно с try/except.

Лабораторные занятия №37-54 (36 часов)

Вычисления по формулам

Вопросы и задания:

1. Составьте программу нахождения медиан треугольника если известны две стороны и угол между ними.
2. Составьте программу выводящую на экран фразу «Тебе N лет» в правильной форме («Тебе пять лет», «Тебе три года», «Тебе 21 год»). N вводится с клавиатуры.
3. Составьте программу вычисления суммы ряда  $S=1+4+9+...+N^2$ . N вводится с клавиатуры.
4. Составьте программу, выводящую на экран квадраты числе от A до B. A и B вводятся с клавиатуры.
5. Составьте программу вывода на экран всех простых делителей числа N.
6. Составьте программу подсчета суммы четных цифр числа.
7. На вход подается целое натуральное число N и N строк вида: <Фамилия> <название предмета> <балл>. Выведите фамилии студентов с самым низким средним баллом.

Пример входных данных:

5

Иванов Математика 5

Иванов Физика 3

Петров Математика 4

Петров История 5

Удальцов Физика 4

Пример выходных данных:

Удальцов 4

Иванов 4

8. Вводится строка, содержащая буквы, целые неотрицательные числа и иные символы. Вывести все числа, которые встречаются в строке. Например, если дана строка "data 48 call 9 read13 blank0a", то на экран выводятся числа 48, 9.

9. Составьте программу подсчета сколько слов в тексте начинается с гласной буквы и сколько с согласной, ответ выведите на экран.

10. Посчитайте сколько слов во введенном тексте заканчивается на звонкую согласную.

11. Подсчитайте сколько раз каждая буква встречается в тексте – регистр не учитывать. Буквы выведите в алфавитном порядке, буквы отсутствующие в тексте не выводить.

Пример входных данных:

Мама мыла раму

Пример выходных данных:

a – 4

л – 1

м – 4

р – 1

у – 1

ы – 1

12. Подсчитайте сколько различных цифр используется при записи числа N. N вводится с клавиатуры.

13. Выделите из текста имена. Именами считаются все конструкции, записанные с большой буквы и следующие за словами Уважаемый/Уважаемая/Господин/Госпожа/Мистер/Мисс/Миссис/Дорогой/Дорогая.

14. Дан список имён (например, ['Маша', 'Петя', 'Андрей Иванович']). Напишите программу, которая вывод на экран последовательность приветствий. Причем к тем, чьё имя записано одним словом, обращается «Привет», а к остальным «Здравствуйте»:

«Привет, Маша!»

«Здравствуйте, Андрей Иванович»

15. Составьте программы выводящую на экран все слова текстового файла длины N. N вводится с клавиатуры.

16. Составьте программу, которая спрашивает с пользователя анкетные данные (ФИО, год рождения, город проживания) и записывает их в файл с названием, состоящим из города и фамилии, например Иванов\_ Самара.txt.

17. Составьте программу, которая построчно читает информацию их файла X, и перезаписывает последнее слово строки в файл Y.

18. Примерное задание: Напишите функцию, проверяющую является ли введенная строка адресом электронной почты (содержит символ @, состоит из букв латинского алфавита, знака подчеркивания и цифр и т.д.)

19. Примерное задание: Напишите рекурсивную функцию Double(n) выводящую на экран строчку вида 0 2 4..n-2 n либо 1 3 5..n-2..n, в зависимости от четности N.

20. Примерное задание: Напишите программу, которая получает на вход количество пар страна-столица N. Далее идет N строк, каждая строка начинается с названия страны, затем идет название столицы. Далее следует запрос «Введите данные», если введен город из списка на экран выводится страна, если страна – её столица, если данных нет в списке выводится соответствующее сообщение.

### **6 семестр, 11 лекций, 18 лабораторных занятий**

#### **Раздел 4. Подключаемые библиотеки**

Лекция №34 (2 часа)

Модули

Вопросы и задания:

Именные и анонимные функции, инструкции «def», «return» и «lambda», функции с переменным числом аргументов, необязательные аргументы. Понятие модуля. Импорт модуля: инструкции «import» и «from». Компиляция. Стандартные модули. Sys, Math, Abs. Примеры работы

Лекции №35-36 (4 часа)

Библиотеки для работы с веб-данными

Вопросы и задания:

Библиотека request. Установка внешних библиотек, модуль pip. Методы библиотеки: post и get. Requests: сбор данных веб-страницы. Библиотека BeautifulSoup: анализ веб-данных, методы find и findall.

Лекция №37-38 (4 часа)

Графический интерфейс: библиотека Tkinter

Вопросы и задания:

Графический интерфейс пользователя (GUI). Работа с графическим интерфейсом. Библиотека Tkinter: формы, виджеты, обработка событий. Менеджеры геометрии.

Лекции №39-40 (4 часа)

Графический интерфейс: библиотека PyGame

Вопросы и задания:

Введение в разработку игра на Python. Модуль pygame.draw – геометрические примитивы. События клавиатуры События мыши. Класс Surface и метод blit(). Модуль pygame.font. Модули pygame.image и pygame.transform. Классы Sprite и Group



Лекция №41-42 (4 часа)

Фреймворк для создания веб-приложений: Flask

Вопросы и задания:

Шаблоны. Формы. База данных. Вход пользователей. Страница профиля и аватары. Unit-тестирование. Полнотекстовый поиск. Поддержка e-mail. Дата и время. Ajax. Отладка, тестирование и профилирование

Лекции №43-44 (4 часа)

Голосовой интерфейс

Вопросы и задания:

Голосовые интерфейсы: примеры, история создания. Навыки для голосового помощника «Алиса». Этапы разработки. Проектирование, публикация и тестирования навыка.

Лабораторные занятия №55-72 (36 часов)

Модули

Вопросы и задания:

1. Напишите программу «Антонимы». Программа должна предлагать уровень сложности. На первом уровне программа предлагает пользователю слово и несколько вариантов антонимов к нему (например, слово «тёплый» и варианты антонимов «мокрый», «пресный», «холодный») за верный ответ начисляется 1 балл. На втором уровне сложности пользователь должен сам вводить антоним, за верный ответ начисляется 3 балла. После каждого ответа пользователю предлагают продолжить игру и выбрать уровень сложности, либо выйти из игры. Ведется учет рекордов.
2. Пользователь вводит адреса двух сайтов. Составьте программу, проверяющую встречаются ли на данных сайтах одинаковые ссылки, и, если встречаются, выводящую их на экран.
3. Напишите программу выводящую суммарную численность обучающихся очно-заочной формы обучения используйте данные со страницы «Образование» (<http://www.pgsga.ru/sveden/education/#study>)
5. Напишите скрипт, в котором с помощью двух блоков радиокнопок можно будет выбрать цвет (один из трех) и размер (один из двух) кнопки. Изменения происходят после каждого нажатия кнопки.
6. Напишите программу, проверяющую правильность решения задач на умножение. На форме располагаются метка (Label) с текстом «x\*y=», однострочное текстовое поле, для ввода ответа, и кнопка «проверить». X и y случайные целые числа от 0 до 100. Ответ должен быть «Верно» или «Увы, Вы не правы».
7. Напишите графическое приложение «Репетитор по английскому». Возможен выбор режима перевод с английского на русский, и наоборот. Регистр игнорируется (то есть, и «Кот», и «кот», и «КОТ» являются верными ответами). Одному слову может соответствовать несколько вариантов перевода. За верное решение с первой попытки начисляется 3 балла, со второй 2 балла, с третьей 1, после неверной третьей попытки программа сама выдает перевод и переходит к следующему слову. На каждом этапе должна быть возможность выхода из программы. По завершении программы выдается количество набранных очков. В программе предполагается запись рекордов.
8. Разработка игры «Астронавт»:
  - Основной цикл.
  - Обработка событий
  - Обновление состояния
  - Среда разработки
  - Движение объектов
9. Создание веб-приложения с помощью Flask:
  - Установка Flask.
  - Создание базового приложения
  - Использование шаблонов HTML
  - Настройка базы данных
  - Расширения Flask
10. Разработка навыка голосового помощника «Угадай художника».
  - Проектирование навыка. Регистрация аккаунта Яндекс.
  - Разворачивание сервиса, поддерживающего протокол взаимодействия с Яндекс.Диалогами (сервис Now, Microsoft Azure, Amazon Web Services, YandexCloudFunctions).
  - Написание программы или функции.
  - Публикация навыка.
  - Тестирование навыка.

7 семестр, 23 лекций, 35 лабораторных занятий

Раздел 5. Объектно-ориентированное программирование

Лекции №45-46 (4 часа)

Элементарные структуры данных

Вопросы и задания:

Базовые принципы, используемые для организации данных. Свойства массивов, связанных списков и строк. Операции для управления структурами данных. Хранение данных в виде объектов переменных размеров.

Лекция №47-48 (4 часа)

Абстрактные типы данных

Вопросы и задания:

Уровни абстракции в вычислительных системах. Понятие класса и объекта. Абстрактные типы данных: стеки и очереди. Права доступа. Общие, частные, защищенные. Наследование прав.

Лекции №49-50 (4 часа)

Рекурсия и деревья

Вопросы и задания:

Рекурсивные алгоритмы. Динамическое программирование. Деревья. Математические свойства бинарных деревьев. Обход дерева. Рекурсивные алгоритмы для бинарных деревьев. Обход графа.

Лекции №51-52 (4 часа)

Методы сортировки

Вопросы и задания:

Элементарные методы сортировки. Быстрая сортировка. Слияние и сортировка слиянием. Очереди с приоритетами и пирамидальная сортировка. Поразрядная сортировка. Специальные методы сортировки.

Лекции №53-54 (4 часа)

Таблицы символов и деревья бинарного поиска

Вопросы и задания:

Абстрактный тип данных таблицы символов. Распределяющий поиск. Последовательный поиск. Бинарный поиск. Деревья бинарного поиска. Характеристики производительности деревьев бинарного поиска. Индексные реализации таблиц символов. Вставка в корень в деревьях бинарного поиска. Реализации других функций АД с помощью BST - ДЕРЕВА.

Лекции №55-56 (4 часа)

Сбалансированные деревья

Вопросы и задания:

Рандомизированные BST-деревья. Скошенные деревья бинарного поиска. Нисходящие 2-3-4-деревья. RB-деревья. Слоеные списки. Характеристики производительности.

Лекции №57-58 (4 часа)

Хэширование

Вопросы и задания:

Хеш-функции. Цепочки переполнения. Линейное опробование. Двойное хэширование. Динамические хеш-таблицы. Перспективы.

Лекции №59-61 (6 часов)

Поразрядный поиск

Вопросы и задания:

Деревья цифрового поиска. Trie-деревья. Patricia-деревья. Многопутевые trie-деревья и TST-деревья. Алгоритмы индексирования текстовых строк.

Лекции №62-64 (6 часов)

Внешний поиск

Вопросы и задания:

Индексно-последовательный доступ. B-деревья. Расширяемое хэширование. Примеры реализации.

Лекции №65-67 (6 часов)

Поиск на графе

Вопросы и задания:

Исследование лабиринта. Поиск в глубину. Функции АД поиска на графе. Свойства лесов DFS. Алгоритмы DFS. Разделимость и двусвязность. Поиск в ширину. Обобщенный поиск на графах. Анализ алгоритмов на графах.

Лабораторные занятия №73-107 (70 часов)

Элементарные структуры данных

Вопросы и задания:

1. Напишите программу сортировки строк трехмерного массива.
2. Напишите программу заполнения двумерного массива значениями 0 или 1: элемент  $a[i][j]$  должен содержать значение 1, если наибольший общий делитель  $i$  и  $j$  равен единице, и значение 0 в остальных случаях.
3. Реализуйте перемножение матриц, представленных мульти списками.
4. Напишите программу для стека магазинного типа, которая считывает любое постфиксное выражение с операциями умножения и сложения целых чисел, затем вычисляет это выражение и выводит полученный результат.
5. Напишите программу реализации очереди FIFO на базе массива. К содержимому очереди относятся все элементы массива, расположенные между индексами head и tail, учитывая возврат к 0 с конца массива. Если индексы head и tail равны, очередь считается пустой, однако если они стали равными в результате выполнения операции put, очередь считается полной.
6. Реализуйте поиск в глубину с использованием стека для графов, которые представлены списками смежности.
7. Реализуйте рекурсивный поиск в глубину для графов, которые представлены списками смежности.
8. Приведите неадаптивную реализацию сортировки выбором, основанную на поиске минимального элемента.
9. Разработайте эффективную реализацию пузырьковой сортировки с минимально возможным числом операторов во внутреннем цикле.
10. Напишите программу, которая решает задачу обобщенного R-путевого разбиения.
11. Напишите программу, которая генерирует случайные 80-байтовые ключи, а потом сортирует их методом поразрядной MSD-сортировки для  $N = 10^3, 10^4, 10^5$  и  $10^6$ . Добавьте возможность вывода общего количества байтов, проверенных в процессе каждой сортировки.
12. Приведите рекурсивную версию нечетно-четного слияния Бэтчера.
13. Создайте сортирующие сети, способные выполнять сортировку 3-упорядоченных и 4-упорядоченных файлов из  $N$  элементов.

14. Напишите рекурсивную программу, которая вычисляет максимальное количество сравнений, требуемых для любого поиска в данном BST-дереве (высоту дерева).
15. Разработайте нерекурсивную реализацию вставки в корень BST-дерева.
16. Напишите программу вычисления вероятности того, что рандомизированная вставка завершается в одном из внутренних узлов заданного дерева, для каждого из узлов на пути поиска.
17. Реализуйте сбалансированные 2-3-4-деревья, используя представление RB-дерева, в котором 3-узлы всегда наклонены вправо. Примечание: это изменение позволяет исключить из внутреннего цикла операции вставить одну битовую проверку.
18. Разработайте функцию хеширования для строковых ключей, основанную на идее одновременной загрузки 4 байтов с последующим выполнением арифметических операций сразу над 32 битами.  
Напишите программу, которая вставляет 105 случайных неотрицательных целых чисел, меньших 106, в таблицу размером 105, используя линейное опробование. Программа должна выводить гистограмму количества проб, использованных для каждого из 103 последовательных вставок.  
Реализуйте операцию удалить для таблицы символов на основе DST-дерева.  
Напишите программу, которая выводит все ключи trie-дерева, имеющие те же начальные  $t$  разрядов, что и заданный ключ.  
Напишите программу для вычисления среднего количества внешних страниц B-дерева порядка  $M$ , построенного  $N$  случайными вставками в первоначально пустое дерево при использовании вероятностного процесса  
Напишите программу для вычисления среднего количества внешних страниц и среднего размера каталога для расширяемой хеш-таблицы, построенной  $N$  случайными вставками в первоначально пустое дерево, при емкости страницы  $M$ .  
Напишите реализацию клиентской функции АДГ графа, которая вызывает указанную клиентом функцию для каждой вершины графа.  
Напишите программу, которая вычисляет представление родительскими ссылками полного дерева DFS (включая внешние узлы) с помощью вектора из  $E$  целых чисел от 0 до  $V-1$ . Указание. Первые  $V$  компонентов этого вектора должны совпадать с компонентами вектора  $st$ , описанного в тексте.

**8 семестр, 23 лекций, 35 лабораторных занятий**

**Раздел 6 Объектно-ориентированное программирование**

Лекции №68-69 (4 часа)

Введение в объектно-ориентированное программирование

Вопросы и задания:

Понятие ООП (объектно-ориентированного программирования). Принципы ООП: абстрагирование, инкапсуляция, ограничение доступа, модульность, иерархичность, наследование, типизация, параллелизм, устойчивость. Этапы разработки программ с использованием ООП.

Лекции №70-71 (4 часа)

Конструкции классов и области видимости

Вопросы и задания:

Понятие класса. Конструкция класса. Обращение к компонентам класса. Область видимости. Локальная область видимости. Глобальные объекты и функции. Объявления и определения.

Лекции №72-73 (4 часа)

Конструктор и деструктор

Вопросы и задания:

Конструктор. Вызов конструктора. Конструктор по умолчанию. Конструктор инициализации. Конструктор копирования. Деструктор. Вложенные объекты.

Лекции №74-75 (4 часа)

Использование указателей в классе

Вопросы и задания:

Указатели на компоненты класса. Варианты взаимодействия между вложенными объектами. Обращение к вложенному объекту через интерфейсные методы владельца.

Лекции №76-77 (4 часа)

Наследование

Вопросы и задания:

Наследование. Иерархия наследования. Определение производного класса. Конструкторы и деструкторы производных классов. Множественное наследование. Виртуальное наследование.

Лекции №78-79 (4 часа)

Полиморфизм

Вопросы и задания:

Перегрузка. Переопределение. Простой (статический) полиморфизм. Полиморфные классы. Переопределение методов в производных классах.

Лекции №80-81 (4 часа)

Виды классов

Вопросы и задания:

Вложенные и локальные классы. Параметризованные классы. Простой (статический) полиморфизм. Полиморфные классы. Переопределение методов в производных классах. Параметризованные классы. Параметризованные функции. Шаблон функции. Контейнерные классы.

Лекции №82-83 (4 часа)

Исключительные ситуации и их обработка

Вопросы и задания:

Исключения. Стандартные библиотеки ASSERT.H и SIGNAL.H. Генерация исключений. Спецификация исключения.  
 Лекции №84-85 (4 часа)

Перезагрузка операций в классах

Вопросы и задания:

Перезагрузка операций в классах. Функции-операторы: бинарные и унарные. Перегружаемые операции. Форматы вызова.  
 Лекции №86-87 (4 часа)

Стандартная библиотека шаблонов STL

Вопросы и задания:

Компоненты стандартной библиотеки STL. Контейнеры STL. Интерфейсы контейнеров STL. Определение ассоциативных контейнеров STL и их конструкторов. Адаптеры контейнеров STL. Очередь. Итераторы STL. Алгоритмы STL.

Лекции № 88-90 (6 часов)

Потоковый ввод и вывод

Вопросы и задания:

Библиотека потоковых классов. Стандартные потоки. Строковые потоки. Файловые потоки. Флаги и манипуляторы. L. Функции проверки состояния чтения из файла.

Лабораторные занятия №108-142 (70 часов)

Основы ООП

Вопросы и задания:

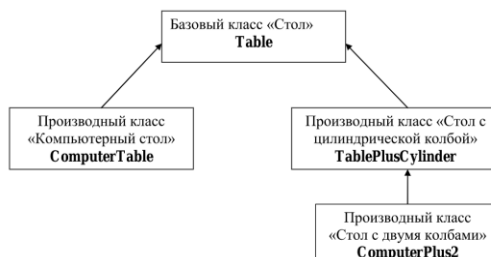
Написать спецификацию для описания класса "транспортное средство для передвижения пассажиров" (не более 10 характеристик и не более 5 методов)

Опишите новый тип Complex – класс комплексных чисел. Создайте интерфейс для ввода и вывода чисел в привычном виде 0,5+5,7i.

Опишите класс, экземпляры которого моделируют какие-либо объекты реального мира. К классу предъявляются следующие требования: как минимум два уровня вложенности, все поля в закрытом разделе, полноценные конструкторы, деструктор, конструктивные методы, отдельный модуль.

Реализовать задачу с несколькими вложенными объектами. Рассмотреть возможность делегирования методов между вложенными объектами.

Создайте иерархию классов вида:



Определите в классах потомках новые методы, вызывайте в методах производного класса методы базового класса.

Реализовать задачу, демонстрирующую применения статических функций и приведения типов.

Предложите иерархию классов с родительским абстрактным классом.

Определить действия виртуальных функций и наследуемой функции.

Для класса square добавьте метод для поворота на определенный угол.

Написать параметризованную функцию, реализующую любой алгоритм сортировки, где параметром является тип элементов массива.

Написать вызов этой функции для целых, вещественных и символьных типов.

Реализовать обработку исключения для заданной иерархии.

Переопределить самостоятельно для класса «Комплексные числа» операции: разность двух комплексных чисел; унарную операцию «--», которая будет отнимать от вещественной и мнимой части по 1 и возвращая измененное число.

Используя шаблоны list и метод sort, организуйте список ваших объектов и выведите информацию о них по возрастанию одного из параметров.

Добавьте в программу проверку ввода числовых данных с экрана.

Напишите метод ввода данных в текстовый файл.

Организируйте ввод данных из текстового файла.

## 5.2. Содержание самостоятельной работы по дисциплине (модулю)

### Содержание обязательной самостоятельной работы по дисциплине

№ п/п	Темы дисциплины	Содержание самостоятельной работы	Продукты деятельности
1	Процедурное программирование	Написание программ	Программы
2	Структуры данных	Написание программ	Программы
3	Мультипарадигмальное программирование	Написание программ	Программы
4	Подключаемые библиотеки	Написание программ	Программы
5	Объектно-ориентированное программирование	Написание программ	Программы

6	Объектно-ориентированное программирование	Письменный отчет по лабораторной работе	Отчет
<b>Содержание самостоятельной работы по дисциплине на выбор</b>			
№ п/п	Темы дисциплины	Содержание самостоятельной работы	Продукты деятельности
1	Процедурное программирование	Решение задач повышенной сложности	Решенная задача
2	Структуры данных	Решение задач повышенной сложности	Решенная задача
3	Мультипарадигмальное программирование	Решение задач повышенной сложности	Решенная задача
4	Подключаемые библиотеки	Решение задач повышенной сложности	Решенная задача
5	Объектно-ориентированное программирование	Решение задач повышенной сложности	Решенная задача
6	Объектно-ориентированное программирование	Подготовка пакета задач с решениями по выбранной теме	Набор задач
<b>5.3. Образовательные технологии</b>			
При организации изучения дисциплины будут использованы следующие образовательные технологии: информационно-коммуникационные технологии, технология организации самостоятельной работы, технология рефлексивного обучения, технология модульного обучения, технология игрового обучения, технологии групповой дискуссии, интерактивные технологии, технология проблемного обучения, технология организации учебно-исследовательской деятельности, технология проектного обучения, технология развития критического мышления.			
<b>5.4. Текущий контроль, промежуточный контроль и промежуточная аттестация</b>			
Балльно-рейтинговая карта дисциплины оформлена как приложение к рабочей программе дисциплины. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине оформлен отдельным документом.			
<b>6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ</b>			
<b>6.1. Рекомендуемая литература</b>			
<b>6.1.1. Основная литература</b>			
	Авторы, составители	Заглавие, ссылка на электронную библиотечную систему	Издательство, год
Л1.1	Седжвик, Р.	Алгоритмы на C++ URL: <a href="https://biblioclub.ru/index.php?page=book_red&amp;id=429164">https://biblioclub.ru/index.php?page=book_red&amp;id=429164</a>	Национальный Открытый Университет «ИНТУИТ», 2016
<b>6.1.2. Дополнительная литература</b>			
	Авторы, составители	Заглавие, ссылка на электронную библиотечную систему	Издательство, год
Л2.1	Корчуганова, М. Р., Иванов, К. С., Бондарева, Л. В.	Объектно-ориентированное программирование на C++: электронное учебное пособие: учебное пособие. URL: <a href="https://biblioclub.ru/index.php?page=book_red&amp;id=481559">https://biblioclub.ru/index.php?page=book_red&amp;id=481559</a>	Кемеровский государственный университет, 2015
<b>6.2 Перечень программного обеспечения</b>			
- Acrobat Reader DC			
- Dr.Web Desktop Security Suite, Dr.Web Server Security Suite			
- GIMP			
- Microsoft Office 365 Pro Plus - subscription license (12 month) (Пакет программ Word, Excel, Access, PowerPoint, Outlook, OneNote, Publisher, Teams, OneDrive, Yammer, Stream, SharePoint Online).			
- Microsoft Windows 10 Education			
- XnView			
- Архиватор 7-Zip			
<b>6.3 Перечень информационных справочных систем, профессиональных баз данных</b>			
- ЭБС «Университетская библиотека онлайн»			
- Базы данных Springer eBooks			
<b>7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)</b>			
7.1	Наименование специального помещения: помещение для самостоятельной работы, Читальный. Оснащенность: ПК-4шт. с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду СГСПУ, Письменный стол-4 шт., Парта-2 шт.		
7.2	Наименование специального помещения: учебная аудитория для проведения занятий лекционного типа, лабораторных занятий, групповых консультаций, индивидуальных консультаций, текущего контроля, промежуточной аттестации, помещение для самостоятельной работы, Педагогический технопарк «Кванториум» им. В. Ф. Волкова "Лаборатория мехатроники и соревновательной робототехники", помещение для хранения и профилактического обслуживания учебного оборудования. Оснащенность: Меловая доска-1шт., Комплект учебной мебели, ноутбук, проекционное оборудование (мультимедийный проектор и экран).		

#### **8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)**

Работа над теоретическим материалом происходит кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометить важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю.

Проработка рабочей программы дисциплины, уделяя особое внимание целям и задачам, структуре и содержанию дисциплины. Конспектирование источников, подготовка ответов к контрольным вопросам, просмотр рекомендуемой литературы, работа с информационными источниками в разных форматах.

Также в процессе изучения дисциплины методические рекомендации могут быть изданы отдельным документом

Балльно-рейтинговая карта дисциплины «Программирование»

Курс 2 Семестр 3

Вид контроля		Минимальное количество баллов	Максимальное количество баллов
<b>Наименование раздела</b> Процедурное программирование			
Текущий контроль по разделу:			
1	Аудиторная работа	16	26
2	Самостоятельная работа (специальные обязательные формы)	10	13
3	Самостоятельная работа (специальные формы на выбор)	3	5
Контрольное мероприятие по разделу			
Промежуточный контроль			
Промежуточная аттестация			
		29	56
		<b>Итого:</b>	<b>100</b>
Виды контроля	Перечень или примеры заданий, критерии оценки и количество баллов	Темы для изучения и образовательные результаты	
<b>Текущий контроль по разделу «Процедурное программирование»</b>			
1	<p>Аудиторная работа</p> <p>Пример заданий                      Лабораторная работа №1 «Вычисление по формулам»                      Пример задания: 1. составьте программу вычисления работы при равномерном поднятии груза массой 2 т. На высоту 50 см                      2. составьте программу вычисления высот треугольника если известны его стороны А и В и угол между ними Q.</p> <p>Критерии оценивания:                      • не решал задачи или решил неправильно – 0 баллов;                      • задачи решены с несущественными ошибками – 1 балл;                      • задачи решены без ошибок – 2 балла.                      Итого – 13x2=26 баллов</p>	<p>Темы:                      1. Вычисления по формулам                      2. Условный оператор IF ... THEN                      3. Логические операции...                      4. Оператор цикла с параметром                      5. Элементарные задачи целочисленной арифметики                      6. Оператор цикла с предусловием                      7. Оператор цикла с постусловием                      8. Организация циклов (различных)                      9. Перечислимый и ограниченный типы данных.                      10. Оператор выбора                      11. Структурирование программ с использованием процедур                      12. Структурирование программ с использованием функций                      13. Рекуррентные соотношения</p> <p>Знает: этапы решения задачи на компьютере; способы формализации алгоритмов на языках программирования высокого уровня; методы процедурного программирования; основные конструкции языка программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени.</p> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>	
2	<p>Самостоятельная работа</p> <p>Индивидуальное домашнее задание (ИДЗ) (x13)</p>	<p>Темы:                      1. Вычисления по формулам</p>	

Направление подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки)  
 Направленность (профиль) «Информатика» и «Дополнительное образование (в области информатики и ИКТ)»  
 Рабочая программа дисциплины «Программирование»

	(обязательные формы)	<p>Составьте программу подсчета простых делителей числа N</p> <p>Составьте программы подсчета всех трехзначных чисел в десятичной записи которых нет одинаковых чисел.</p> <ul style="list-style-type: none"> <li>решены все задачи ИДЗ – 0,5 балла;</li> <li>решения задач оформлены развернуто, и представлены в срок – 0,5 балл;</li> </ul> <p>Итого – 13x1=13 баллов</p>	<ol style="list-style-type: none"> <li>Условный оператор IF ... THEN</li> <li>Логические операции...</li> <li>Оператор цикла с параметром</li> <li>Элементарные задачи целочисленной арифметики</li> <li>Оператор цикла с предусловием</li> <li>Оператор цикла с постусловием</li> <li>Организация циклов (различных)</li> <li>Перечислимый и ограниченный типы данных.</li> <li>Оператор выбора</li> <li>Структурирование программ с использованием процедур</li> <li>Структурирование программ с использованием функций</li> <li>Рекуррентные соотношения</li> </ol> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>
3	Самостоятельная работа (на выбор)	<p>Решение задач повышенной сложности.</p> <ul style="list-style-type: none"> <li>решены все задачи – 3 балла;</li> <li>решения задач оформлены развернуто, в соответствии с требованиями преподавателя – 2 балла.</li> </ul> <p>Итого – 5 баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>Логические операции...</li> <li>Оператор цикла с параметром</li> <li>Элементарные задачи целочисленной арифметики</li> <li>Оператор цикла с предусловием</li> <li>Оператор цикла с постусловием</li> <li>Перечислимый и ограниченный типы данных.</li> <li>Оператор выбора</li> <li>Структурирование программ с использованием процедур</li> <li>Структурирование программ с использованием функций</li> <li>Рекуррентные соотношения</li> </ol> <p>Образовательные результаты:</p> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>
	Контрольное мероприятие по разделу		
	Промежуточный контроль (количество баллов)	29-56 баллов	
	Промежуточная аттестация	Представлены в фонде оценочных средств для промежуточной аттестации по дисциплине	



Курс 2 Семестр 4

Вид контроля		Минимальное количество баллов	Максимальное количество баллов
<b>Наименование раздела</b> Процедурное программирование			
Текущий контроль по разделу:			
1	Аудиторная работа	13	26
2	Самостоятельная работа (специальные обязательные формы)	5	10
3	Самостоятельная работа (специальные формы на выбор)	2	4
Контрольное мероприятие по разделу			
Промежуточный контроль		26	40
Промежуточная аттестация		30	60
Итого:		<b>56</b>	<b>100</b>

Виды контроля	Перечень или примеры заданий, критерии оценки и количество баллов	Темы для изучения и образовательные результаты
<b>Текущий контроль по разделу «Структуры данных»</b>		
1	<p>Аудиторная работа</p> <p>Пример заданий:                      Лабораторная работа №1 «Структуры данных: двумерные массивы»                      1. Составьте программу нахождения чаще всего встречающегося элемента массива C(m,n)                      2. Составьте программу, позволяющую выяснить есть ли в массиве столбцы, в которых все элементы различны.</p> <p>Критерии оценивания:                      • не решал задачи или решил неправильно – 0 баллов;                      • задачи решены с несущественными ошибками – 1 балл;                      • задачи решены без ошибок – 2 балла.                      Итого – 13x2=26 баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>1. Структуры данных: строки</li> <li>2. Структуры данных: одномерные массивы</li> <li>3. Структуры данных: двумерные массивы</li> <li>4. Структура данных: множества.</li> <li>5. Структура данных: файлы</li> <li>6. Структура данных: записи.</li> <li>7. Графические примитивы</li> <li>8. График функций в декартовой системе координат</li> <li>9. График функций в полярной системе координат</li> <li>10. Построение кривых, заданных в параметрической форме.</li> <li>11. Создание рисунков.</li> <li>12. Имитация движения с помощью видеостраниц.</li> <li>13. Избранные задачи.</li> </ol> <p>Знает: этапы решения задачи на компьютере; способы формализации алгоритмов на языках программирования высокого уровня; методы процедурного программирования; основные конструкции языка программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени.</p> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при</p>

Направление подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки)  
 Направленность (профиль) «Информатика» и «Дополнительное образование (в области информатики и ИКТ)»  
 Рабочая программа дисциплины «Программирование»

			некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.
2	Самостоятельная работа (обязательные формы)	<p>Индивидуальное домашнее задание (ИДЗ) (x5)</p> <ol style="list-style-type: none"> <li>Используя модуль <code>graph</code> или <code>str</code> языка программирования Паскаль написать программу "Змейка".</li> <li>В файле <code>input.txt</code> записаны числа (каждое — с новой строки), их количество не превышает 100. Необходимо отсортировать их по возрастанию и записать в файл <code>output.txt</code>.</li> </ol> <ul style="list-style-type: none"> <li>решены все задачи ИДЗ – 1 балла;</li> <li>решения задач оформлены развернуто, в соответствии с требованиями преподавателя – 0,5 балла;</li> <li>отчет представлен преподавателю в установленные сроки – 0,5 балла.</li> </ul> <p>Итого – <math>5 \times 2 = 10</math> баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>Строковый тип данных</li> <li>Массивы</li> <li>Записи</li> <li>Файлы</li> <li>Графика</li> </ol> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>
3	Самостоятельная работа (на выбор)	<p>Решение задач повышенной сложности.</p> <ul style="list-style-type: none"> <li>решены все задачи – 3 балла;</li> <li>решения задач оформлены развернуто, в соответствии с требованиями преподавателя – 2 балла.</li> </ul> <p>Итого – 5 баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>Структуры данных: одномерные массивы</li> <li>Структуры данных: двумерные массивы</li> <li>Структура данных: множества.</li> <li>Структура данных: файлы</li> <li>Структура данных: записи. Образовательные результаты:</li> </ol> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>
Контрольное мероприятие по разделу			
Промежуточный контроль (количество баллов)		30-60 баллов	
Промежуточная аттестация		Представлены в фонде оценочных средств для промежуточной аттестации по дисциплине	

Курс 3 Семестр 5

Вид контроля		Минимальное количество баллов	Максимальное количество баллов
<b>Наименование раздела Мультипарадигмальное программирование</b>			
Текущий контроль по разделу:			
1	Аудиторная работа	18	24
2	Самостоятельная работа (специальные обязательные формы)	6	12
3	Самостоятельная работа (специальные формы на выбор)	2	4
Контрольное мероприятие по разделу			
Промежуточный контроль		26	40
Промежуточная аттестация		30	60
Итого:		<b>56</b>	<b>100</b>

Виды контроля	Перечень или примеры заданий, критерии оценки и количество баллов	Темы для изучения и образовательные результаты
<b>Текущий контроль по разделу «Мультипарадигмальное программирование»</b>		
1	<p>Аудиторная работа</p> <p>Пример заданий:                      Лабораторная работа №9 «Работа с файлами»                      1. Составьте программу, которая выводит на экран слова текстового файла, начинающиеся с буквы, введённой пользователем, в алфавитном порядке.                      2. Составьте программу, позволяющую выяснить есть ли в текстовом файле одинаковые строки</p> <p>Критерии оценивания:                      • не решал задачи или решил неправильно – 0 баллов;                      • задачи решены с несущественными ошибками – 1 балл;                      • задачи решены без ошибок – 2 балла.                      Итого – 12x2=24 баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>1. Вычисления по формулам</li> <li>2. Условный оператор «IF ... ELIF ELSE»</li> <li>3. Циклические алгоритмы, инструкция «FOR»</li> <li>4. Элементарные задачи целочисленной арифметики</li> <li>5. Процедуры обработки списков</li> <li>6. Процедуры обработки слов</li> <li>7. Процедуры обработки множеств</li> <li>8. Кортежи</li> <li>9. Работа с файлами</li> <li>10. Структурирование программ с использованием функций</li> <li>11. Рекуррентные соотношения</li> <li>12. Словари</li> </ol> <p>Знает: этапы решения задачи на компьютере; способы формализации алгоритмов на языках программирования высокого уровня; методы мультипарадигмального программирования; основные конструкции языка программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени.</p> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи;</p>

			составлять систему тестов для автоматизированной проверки корректности программы.
2	Самостоятельная работа (обязательные формы)	<p>Индивидуальное домашнее задание (ИДЗ) (х6)                  Напишите программу, которая имитирует игру «анаграмма», аналогичную разобранный на лекции, но с учетом слов, содержащих повторяющиеся буквы. Кроме того, в программе должен вестись учет рекордов. Пользователь? набравший больше всего очков вводит своё имя, и программа его запоминает. Систему начисления очков разработайте самостоятельно.</p> <ul style="list-style-type: none"> <li>решены все задачи ИДЗ – 1 балла;</li> <li>решения задач оформлены развернуто, в соответствии с требованиями преподавателя – 0,5 балла;</li> <li>отчет представлен преподавателю в установленные сроки – 0,5 балла.</li> </ul> <p>Итого – 6х2=12 баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>Процедуры обработки списков</li> <li>Процедуры обработки слов</li> <li>Процедуры обработки множеств</li> <li>Работа с файлами</li> <li>Структурирование программ с использованием функций</li> <li>Рекуррентные соотношения</li> </ol> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>
3	Самостоятельная работа (на выбор)	<p>Решение задач повышенной сложности.</p> <ul style="list-style-type: none"> <li>решены все задачи – 3 балла;</li> <li>решения задач оформлены развернуто, в соответствии с требованиями преподавателя – 1 балла.</li> </ul> <p>Итого – 4 балла</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>Структурирование программ с использованием функций</li> <li>Рекуррентные соотношения</li> <li>Словари</li> </ol> <p>Образовательные результаты:</p> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>
Контрольное мероприятие по разделу			
Промежуточный контроль (количество баллов)		30-60 баллов	
Промежуточная аттестация		Представлены в фонде оценочных средств для промежуточной аттестации по дисциплине	

Курс 3 Семестр 6

Вид контроля		Минимальное количество баллов	Максимальное количество баллов
<b>Наименование раздела Подключаемые библиотеки</b>			
Текущий контроль по разделу:			
1	Аудиторная работа	18	24
2	Самостоятельная работа (специальные обязательные формы)	6	12
3	Самостоятельная работа (специальные формы на выбор)	2	4
Контрольное мероприятие по разделу			
Промежуточный контроль		26	40
Промежуточная аттестация		30	60
Итого:		<b>56</b>	<b>100</b>

Виды контроля	Перечень или примеры заданий, критерии оценки и количество баллов	Темы для изучения и образовательные результаты
<b>Текущий контроль по разделу «Подключаемые библиотеки»</b>		
1	<p>Аудиторная работа</p> <p>Пример заданий:                      Лабораторная работа №2 «Библиотеки для работы с веб-данными»</p> <ol style="list-style-type: none"> <li>Пользователь вводит адреса двух сайтов. Составьте программу, проверяющую встречаются ли на данных сайтах одинаковые ссылки, и, если встречаются, выводящую их на экран.</li> <li>Напишите программу выводящую суммарную численность обучающихся очно-заочной формы обучения используйте данные со страницы «Образование» (<a href="http://www.pgsga.ru/sveden/education/#study">http://www.pgsga.ru/sveden/education/#study</a>)</li> </ol> <p>Критерии оценивания:</p> <ul style="list-style-type: none"> <li>не решал задачи или решил неправильно – 0 баллов;</li> <li>задачи решены с несущественными ошибками либо отсутствуют комментарии – 3 балла;</li> <li>задачи решены оптимальным методом, содержат комментарии – 4 балла.</li> </ul> <p>Итого – 6x4=24 баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>Модули</li> <li>Библиотеки для работы с веб-данными</li> <li>Графический интерфейс: библиотека Tkinter</li> <li>Графический интерфейс: библиотека PyGame</li> <li>Фреймворк для создания веб-приложений: Flask</li> <li>Голосовой интерфейс</li> </ol> <p>Знает: этапы решения задачи на компьютере; способы формализации алгоритмов на языках программирования высокого уровня; методы мультипарадигмального программирования; основные конструкции языка программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени.</p> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>
2	<p>Самостоятельная работа (обязательные формы)</p> <p>Индивидуальное домашнее задание (ИДЗ) (x6)</p> <ol style="list-style-type: none"> <li>Выведите на экран заголовки всех страниц, на которые ссылается данный сайт в алфавитном порядке. Заголовок сайта располагается между тегами &lt;title&gt; &lt;/title&gt;..</li> </ol>	<p>Темы:</p> <ol style="list-style-type: none"> <li>Модули</li> <li>Библиотеки для работы с веб-данными</li> <li>Графический интерфейс: библиотека Tkinter</li> <li>Графический интерфейс: библиотека PyGame</li> </ol>

Направление подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки)  
 Направленность (профиль) «Информатика» и «Дополнительное образование (в области информатики и ИКТ)»  
 Рабочая программа дисциплины «Программирование»

		<ul style="list-style-type: none"> <li>решены все задачи ИДЗ – 1 балла;</li> <li>решения задач оформлены развернуто, в соответствии с требованиями преподавателя – 0,5 балла;</li> <li>отчет представлен преподавателю в установленные сроки – 0,5 балла.</li> </ul> Итого – 6x2=12 баллов	5. Фреймворк для создания веб-приложений: Flask 6. Голосовой интерфейс Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.
3	Самостоятельная работа (на выбор)	Решение задач повышенной сложности. <ul style="list-style-type: none"> <li>решены все задачи – 3 балла;</li> <li>решения задач оформлены развернуто, в соответствии с требованиями преподавателя – 1 балла.</li> </ul> Итого – 4 балла	Темы: <ol style="list-style-type: none"> <li>Модули</li> <li>Библиотеки для работы с веб-данными</li> <li>Графический интерфейс: библиотека Tkinter</li> </ol> Образовательные результаты: Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.
	Контрольное мероприятие по разделу		
	Промежуточный контроль (количество баллов)	30-60 баллов	
	Промежуточная аттестация	Представлены в фонде оценочных средств для промежуточной аттестации по дисциплине	

Курс 4 Семестр 7

Вид контроля		Минимальное количество баллов	Максимальное количество баллов
<b>Наименование раздела Объектно-ориентированное программирование</b>			
Текущий контроль по разделу:			
1	Аудиторная работа	18	24
2	Самостоятельная работа (специальные обязательные формы)	6	12
3	Самостоятельная работа (специальные формы на выбор)	2	4
Контрольное мероприятие по разделу			
Промежуточный контроль		26	40
Промежуточная аттестация		30	60
Итого:		<b>56</b>	<b>100</b>

Виды контроля	Перечень или примеры заданий, критерии оценки и количество баллов	Темы для изучения и образовательные результаты
<b>Текущий контроль по разделу «Объектно-ориентированное программирование»</b>		
1	<p>Аудиторная работа</p> <p>Пример заданий:                      Лабораторная работа №11 «Внешний поиск»</p> <p>1. Напишите программу для вычисления среднего количества внешних страниц В-дерева порядка М, построенного N случайными вставками в первоначально пустое дерево при использовании вероятностного процесса</p> <p>2. Напишите программу для вычисления среднего количества внешних страниц и среднего размера каталога для расширяемой хеш-таблицы, построенной N случайными вставками в первоначально пустое дерево, при емкости страницы М.</p> <p>Критерии оценивания:</p> <ul style="list-style-type: none"> <li>• не решал задачи или решил неправильно – 0 баллов;</li> <li>• задачи решены с несущественными ошибками – 1 балл;</li> <li>• задачи решены без ошибок – 2 балла.</li> </ul> <p>Итого – 12x2=24 баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>1. Элементарные структуры данных</li> <li>2. Абстрактные типы данных</li> <li>3. Рекурсия и деревья</li> <li>4. Элементарные методы сортировки</li> <li>5. Поразрядная сортировка</li> <li>6. Специальные методы сортировки</li> <li>7. Таблицы символов и деревья бинарного поиска</li> <li>8. Сбалансированные деревья</li> <li>9. Хэширование</li> <li>10. Поразрядный поиск</li> <li>11. Внешний поиск</li> <li>12. Поиск на графе</li> </ol> <p>Знает: этапы решения задачи на компьютере; способы формализации алгоритмов на языках программирования высокого уровня; методы объектно-ориентированного программирования; основные конструкции языка программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени.</p> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>

Направление подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки)  
 Направленность (профиль) «Информатика» и «Дополнительное образование (в области информатики и ИКТ)»  
 Рабочая программа дисциплины «Программирование»

2	Самостоятельная работа (обязательные формы)	<p>Индивидуальное домашнее задание (ИДЗ) (x12)</p> <ol style="list-style-type: none"> <li>Создайте нерекурсивную реализацию алгоритма Евклида.</li> <li>Напишите рекурсивную программу для вычисления инфиксных выражений, в которых операнды всегда заключены в круглые скобки.                     <ul style="list-style-type: none"> <li>решены все задачи ИДЗ – 0,5 балла;</li> <li>решения задач оформлены развернуто, в соответствии с требованиями преподавателя – 0,5 балла.</li> </ul> </li> </ol> <p>Итого – 12x1=12 баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>Элементарные структуры данных</li> <li>Абстрактные типы данных</li> <li>Рекурсия и деревья</li> <li>Элементарные методы сортировки</li> <li>Поразрядная сортировка</li> <li>Специальные методы сортировки</li> <li>Таблицы символов и деревья бинарного поиска</li> <li>Сбалансированные деревья</li> <li>Хэширование</li> <li>Поразрядный поиск</li> <li>Внешний поиск</li> <li>Поиск на графе</li> </ol> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>
3	Самостоятельная работа (на выбор)	<p>Решение задач повышенной сложности.</p> <ul style="list-style-type: none"> <li>решены все задачи – 3 балла;</li> <li>решения задач оформлены развернуто, в соответствии с требованиями преподавателя – 3 балла.</li> </ul> <p>Итого – 6 баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>Элементарные методы сортировки</li> <li>Сбалансированные деревья</li> <li>Хэширование</li> <li>Поразрядный поиск</li> <li>Внешний поиск</li> <li>Поиск на графе</li> </ol> <p>Образовательные результаты:</p> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>
Контрольное мероприятие по разделу			
Промежуточный контроль (количество баллов)	30-60 баллов		
Промежуточная аттестация	Представлены в фонде оценочных средств для промежуточной аттестации по дисциплине		



Курс 4 Семестр 8

Вид контроля		Минимальное количество баллов	Максимальное количество баллов
<b>Наименование раздела Объектно-ориентированное программирование</b>			
Текущий контроль по разделу:			
1	Аудиторная работа	19	26
2	Самостоятельная работа (специальные обязательные формы)	7	13
3	Самостоятельная работа (специальные формы на выбор)	0	1
Контрольное мероприятие по разделу			
Промежуточный контроль			
Промежуточная аттестация			
Итого:		<b>56</b>	<b>100</b>

Виды контроля	Перечень или примеры заданий, критерии оценки и количество баллов	Темы для изучения и образовательные результаты
<b>Текущий контроль по разделу «Объектно-ориентированное программирование»</b>		
1	<p>Аудиторная работа</p> <p>1. Пример заданий:                      Лабораторная работа №7 «Виды классов»                      1. редложите иерархию классов с родительским абстрактным классом.                      2. пределить действия виртуальных функций и наследуемой функции.                      3. ля класса square добавьте метод для поворота на определенный угол.</p> <p>Критерии оценивания:                      • не решал задачи или решил неправильно – 0 баллов;                      • задачи решены с несущественными ошибками – 1 балл;                      • задачи решены без ошибок – 2 балла.                      Итого – 13x2=26 баллов</p>	<p>Темы:                      2. Основы ООП                      3. Конструкции классов и области видимости                      4. Конструктор и деструктор                      5. Использование указателей в классе                      6. Наследование                      7. Полиморфизм                      8. Виды классов                      9. Параметризованные функции                      10. Исключительные ситуации и их обработка                      11. Перегрузка операций в классах                      12. Стандартная библиотека шаблонов STL                      13. Поточный ввод и вывод                      14. Поточный ввод и вывод</p> <p>Знает: этапы решения задачи на компьютере; способы формализации алгоритмов на языках программирования высокого уровня; методы объектно-ориентированного программирования; основные конструкции языка программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени.                      Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>

Направление подготовки 44.03.05 Педагогическое образование (с двумя профилями подготовки)  
 Направленность (профиль) «Информатика» и «Дополнительное образование (в области информатики и ИКТ)»  
 Рабочая программа дисциплины «Программирование»

2	Самостоятельная работа (обязательные формы)	<p>Подготовлены текстовые отчеты по заданиям лабораторных работ</p> <p>Критерии оценивания:</p> <ul style="list-style-type: none"> <li>• Отчеты содержат результаты выполнения всех заданий лабораторных работ.</li> <li>• Отчет выполнен и загружен в систему в установленные сроки</li> </ul> <p>обучением в установленные сроки. Каждый критерий оценивается в 0,5 балла. Итого <math>0,5 \times 2 \times 13 = 13</math> баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>1. Основы ООП</li> <li>2. Конструкции классов и области видимости</li> <li>3. Конструктор и деструктор</li> <li>4. Использование указателей в классе</li> <li>5. Наследование</li> <li>6. Полиморфизм</li> <li>7. Виды классов</li> <li>8. Параметризованные функции</li> <li>9. Исключительные ситуации и их обработка</li> <li>10. Перегрузка операций в классах</li> <li>11. Стандартная библиотека шаблонов STL</li> <li>12. Поточный ввод и вывод</li> <li>13. Поточный ввод и вывод</li> </ol> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>
3	Самостоятельная работа (на выбор)	<p>Подготовлен пакет задач с решениями по выбранной теме.</p> <ul style="list-style-type: none"> <li>• приведено не менее 5 задач – 0,5 балла;</li> <li>• приведено 10 задач – 1 балл.</li> </ul> <p>Итого – 1 балл</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>1. Параметризованные функции</li> <li>2. Исключительные ситуации и их обработка</li> <li>3. Перегрузка операций в классах</li> <li>4. Стандартная библиотека шаблонов STL</li> </ol> <p>Образовательные результаты:</p> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы; выполнять оценку сложности алгоритмов, проводить анализ и оценивание полученных результатов..</p>
Контрольное мероприятие по разделу			
Промежуточный контроль (количество баллов)		30-60 баллов	
Промежуточная аттестация		Представлены в фонде оценочных средств для промежуточной аттестации по дисциплине	

Курс 5 Семестр 9

Вид контроля		Минимальное количество баллов	Максимальное количество баллов
<b>Наименование раздела Функциональное программирование</b>			
Текущий контроль по разделу:			
1	Аудиторная работа	19	26
2	Самостоятельная работа (специальные обязательные формы)	7	13
3	Самостоятельная работа (специальные формы на выбор)	0	1
Контрольное мероприятие по разделу			
Промежуточный контроль		26	40
Промежуточная аттестация		30	60
Итого:		<b>56</b>	<b>100</b>

Виды контроля	Перечень или примеры заданий, критерии оценки и количество баллов	Темы для изучения и образовательные результаты
<b>Текущий контроль по разделу «Функциональное программирование»</b>		
1	<p>Аудиторная работа</p> <p>Пример заданий:                      Лабораторная работа №7 «Виды классов»                      4. Предложите иерархию классов с родительским абстрактным классом.                      5. Определить действия виртуальных функций и наследуемой функции.                      6. Для класса square добавьте метод для поворота на определенный угол.</p> <p>Критерии оценивания:                      • не решал задачи или решил неправильно – 0 баллов;                      • задачи решены с несущественными ошибками – 1 балл;                      • задачи решены без ошибок – 2 балла.                      Итого – 13x2=26 баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>1. Черепашья графика (бескоординатный подход)</li> <li>2. Процедуры с параметром</li> <li>3. Работа с формами</li> <li>4. Создание проектов</li> <li>5. Вычисление в режиме непосредственного диалога</li> <li>6. Элементарные вычислительные процедуры</li> <li>7. Процедуры с ветвлением</li> <li>8. Задачи целочисленной арифметики</li> <li>9. Процедуры обработки слов: элементарные задачи</li> <li>10. Процедуры обработки слов</li> <li>11. Система координат</li> <li>12. Процедуры обработки числовых списков</li> <li>13. Разработка проекта учебного назначения</li> </ol> <p>Знает: этапы решения задачи на компьютере; способы формализации алгоритмов на языках программирования высокого уровня; методы функционального программирования; основные конструкции языка программирования; основные типы данных и операторы; приемы оптимизации алгоритмов по памяти и времени.                      Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>

2	Самостоятельная работа (обязательные формы)	<p>Подготовлены текстовые отчеты по заданиям лабораторных работ</p> <p>Критерии оценивания:</p> <ul style="list-style-type: none"> <li>• Отчеты содержат результаты выполнения всех заданий лабораторных работ.</li> <li>• Отчет выполнен и загружен в систему в установленные сроки</li> </ul> <p>обучением в установленные сроки. Каждый критерий оценивается в 0,5 балла. Итого <math>-0,5 \times 2 \times 13 = 13</math> баллов</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>1. Черепашня графика (бескоординатный подход)</li> <li>2. Процедуры с параметром</li> <li>3. Работа с формами</li> <li>4. Создание проектов</li> <li>5. Вычисление в режиме непосредственного диалога</li> <li>6. Элементарные вычислительные процедуры</li> <li>7. Процедуры с ветвлением</li> <li>8. Задачи целочисленной арифметики</li> <li>9. Процедуры обработки слов: элементарные задачи</li> <li>10. Процедуры обработки слов</li> <li>11. Система координат</li> <li>12. Процедуры обработки числовых списков</li> <li>13. Разработка проекта учебного назначения</li> </ol> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы.</p>
3	Самостоятельная работа (на выбор)	<p>Подготовлен пакет задач с решениями по выбранной теме.</p> <ul style="list-style-type: none"> <li>• приведено не менее 5 задач – 0,5 балла;</li> <li>• приведено 10 задач – 1 балл.</li> </ul> <p>Итого – 1 балл</p>	<p>Темы:</p> <ol style="list-style-type: none"> <li>1. Процедуры обработки числовых списков</li> <li>2. Разработка проекта учебного назначения</li> </ol> <p>Образовательные результаты:</p> <p>Умеет: анализировать задачу, выделяя её базовые составляющие; осуществлять декомпозицию задачи; осуществлять постановку задачи; анализировать условие и определять оптимальный метод решения поставленной задачи; строить математическую модель; составлять алгоритм решения задачи; записывать разрабатываемые алгоритмы на языках программирования высокого уровня; комментировать синтаксические и семантические ошибки, возникающие при некорректном выполнении программы; отлаживать и тестировать задачи; составлять систему тестов для автоматизированной проверки корректности программы; выполнять оценку сложности алгоритмов, проводить анализ и оценивание полученных результатов..</p>
Контрольное мероприятие по разделу			
Промежуточный контроль (количество баллов)		30-60 баллов	
Промежуточная аттестация		Представлены в фонде оценочных средств для промежуточной аттестации по дисциплине	