Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Кубанский государственный университет»
Факультет компьютерных технологий и прикладной математики



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.Б.11 ПРОГРАММИРОВАНИЕ

Направление подготовки/специальность 09.03.03 Прикладная информатика

Направленность (профиль) / специализация Прикладная информатика в экономике

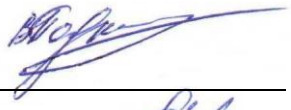
Программа подготовки академическая

Форма обучения очная

Квалификация (степень) выпускника бакалавр

Краснодар 2018

Рабочая программа дисциплины «ПРОГРАММИРОВАНИЕ» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.03 Прикладная информатика

Программу составил(и):
В.В. Подколзин, к.ф.-м.н. 

А.В. Уварова, ст. преподаватель 

Рабочая программа дисциплины Программирование утверждена на заседании кафедры информационных технологий протокол № 13 «7» апреля 2018г.
И.о. заведующего кафедрой (разработчика) Подколзин В.В.



Рабочая программа обсуждена на заседании кафедры прикладной математики протокол № 7 от 18 апреля 2018г.
Заведующий кафедрой (выпускающей) Уртенов М.Х. 
фамилия, инициалы подпись

Утверждена на заседании учебно-методической комиссии факультета Компьютерных технологий и прикладной математики
протокол № 1 от 20 апреля 2018г.
Председатель УМК факультета Малыхин К.В. 
фамилия, инициалы подпись

Рецензенты:

Бегларян Маргарита Евгеньевна, кандидат физико-математических наук, заведующий кафедрой СГЕНД СКФ ФГБОУ ВО «РГУП»

Колотий Александр Дмитриевич, кандидат физико-математических наук, доцент кафедры прикладной математики КубГУ

1 Цели и задачи изучения дисциплины.

1.1 Цель освоения дисциплины.

Цель дисциплины «Программирование» — изучение методов программирования для овладения знаниями в области технологии программирования; подготовка к осознанному использованию, как языков программирования, так и методов программирования.

Воспитательной целью дисциплины является формирование у студентов научного, творческого подхода к освоению технологий, методов и средств производства программного обеспечения.

1.2 Задачи дисциплины.

– знакомство с понятием информация и информационный процесс;

– знакомство с архитектурой ЭВМ, изучение систем счисления и способов представления данных в памяти ПК.

– обучение разработке алгоритмов на основе структурного подхода;

–закрепление навыков алгоритмизации и программирования на основе изучения языка программирования Pascal;

– знакомство с основными структурами данных и типовыми методами обработки этих структур;

– изучение рекурсивных методов и алгоритмов;

Содержательное наполнение дисциплины обусловлено общими задачами в подготовке бакалавра.

1.3 Место дисциплины в структуре образовательной программы.

Дисциплина «Программирование» относится к базовой части Блока 1 "Дисциплины (модули)" учебного плана.

Дисциплина «Программирование» является логически и содержательно-методически связана с такими дисциплинами как ««Программирование в MSOffice», «Дискретные математические системы». Данная дисциплина позволяет заложить основу для изучения программистских дисциплин профессионального цикла.

Входными знаниями для освоения данной дисциплины являются знания, умения и опыт, накопленный студентами в процессе подготовки ЕГЭ и изучения дисциплины «Информатика и ИКТ» в рамках обучения в школе.

1.4 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.

Изучение данной учебной дисциплины направлено на формирование у обучающихся общепрофессиональных компетенций (ОПК)

1. Знать основные естественнонаучные законы хранения и переработки информации;
2. Знать методы и базовые алгоритмы обработки информационных структур;
3. Знать основы концепций, синтаксической и семантической организации, методов использования современных языков программирования и современных информационно-коммуникационных технологий;
4. Уметь составлять и контролировать план выполняемой работы по разработке программ, планировать необходимые для выполнения работы ресурсы, оценивать результаты собственной работы;

5. Уметь применять в профессиональной деятельности основные законы естественнонаучных дисциплин;
6. Владеть современными информационно-коммуникационными технологиями в программировании;

7. Владеть способностью квалифицированно применять в профессиональной деятельности современные информационно-коммуникационные технологии;

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			Знать	**Уметь**	**Владеть**
1.	ОПК-3	способностью использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности	**1,2, 3**	**4, 5**	**6, 7**

2. Структура и содержание дисциплины.

2.1 Распределение трудоёмкости дисциплины по видам работ.

Общая трудоёмкость дисциплины составляет 13 зач.ед. (468 часов), их распределение по видам работ представлено в таблице

(для студентов ОФО).

Вид учебной работы	Всего часов	Семестры (часы)			
		1	2		
Контактная работа, в том числе:					
Аудиторные занятия (всего):					
Занятия лекционного типа	66	34	32	-	-
Лабораторные занятия	106	54	52	-	-
Занятия семинарского типа (семинары, практические занятия)	-	-	-	-	-
	-	-	-	-	-
Иная контактная работа:					
Контроль самостоятельной работы (КСР)	20	10	10		
Промежуточная аттестация (ИКР)	1	0,5	0,5		
Самостоятельная работа, в том числе:					
Курсовая работа	-	-	-	-	-
Проработка учебного (теоретического) материала	100	60	40	-	-
Выполнение индивидуальных заданий	90	50	40	-	-
Реферат	-	-	-	-	-
Подготовка к текущему контролю	13,6	7,8	5,8	-	-

Контроль:						
Подготовка к экзамену		71,4	35,7	35,7		
Общая трудоемкость	**час.**	**468**	**252**	**216**	**-**	**-**
	в том числе контактная работа	**193**	**98,5**	**94,5**		
	зач. Ед	**13**	**7**	**6**		

2.2 Структура дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы дисциплины, изучаемые в 1 семестре

№	Наименование разделов	Всего	Количество часов				
			Аудиторная работа			Внеаудиторная работа	
			Л	ПЗ	ЛР	СРС	контроль
1	2	3	4	5	6	7	
1.	Информация и информационные процессы	23	2		2	12	7
2.	Структура Фон-Неймановской архитектуры	35	6		2	20	7
3.	Теоретические основы компьютерной обработки данных	63	10		16	30	7
4.	Базовые типы данных и методы обработки информации	103	16		32	48	7
5.	Обзор изученного материала и прием зачета	17,5			2	7,8	7,7
6.	Контроль самостоятельной работы (КСР)	10					
7.	Промежуточная аттестация (ИКР)	0,5					
	Итого:	252	34		54	117,8	35,7

Разделы дисциплины, изучаемые во2 семестре

№	Наименование разделов	Всего	Количество часов				
			Аудиторная работа			Внеаудиторная работа	
			Л	ПЗ	ЛР	СРС	контроль
1	2	3	4	5	6	7	8
1.	Рекурсия в программировании	36	6		6	14	10
2.	Структуры данных и основные алгоритмы их обработки	74	16		18	30	10
3.	Объектно-ориентированное программирование	82	10		26	36	10
4.	Обзор изученного материала и прием зачета	13,5			2	5,8	5,7
5.	Контроль самостоятельной работы (КСР)	10					

6.	Промежуточная аттестация (ИКР)	0,5					
	Итого:	216	32		52	85,8	35,7

Примечание: Л – лекции, ПЗ – практические занятия / семинары,ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов дисциплины:

2.3.1 Занятия лекционного типа.

№	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
1.	Информация и информационные процессы	Понятия данные, информация, знания; методы представления, получения, передачи информации; количество и единицы измерения информации; информационные процессы и информационные системы; двоичное кодирование информации, системы счисления.	Индивидуальные задачи коллоквиум
2.	Структура Фон-Неймановской архитектуры	Структура компьютера; назначение основных функциональных элементов; структура основной памяти, понятие бита, байта, адреса.	Индивидуальные задачи коллоквиум
3.	Теоретические основы компьютерной обработки данных	Понятие алгоритма, свойства алгоритмов; компиляция и интерпретация; основные функции языков программирования; основные требования, предъявляемые к языкам программирования; основные аспекты языков программирования; метаязык Бэкуса-Наура, язык синтаксических диаграмм; типы ошибок.	Индивидуальные задачи коллоквиум
4.	Базовые типы данных и методы обработки информации	Понятие типа; структура типов языка С++; описание переменных и констант; целочисленные типы; вещественные типы; арифметические выражения; классификация операторов языка С++; логический тип; оператор присваивания; условные оператор; операторы цикла; оператор выбора; интервальные типы; регулярные типы; строковый тип; подпрограммы.	Индивидуальные задачи коллоквиум
5.	Рекурсия в программировании	Рекурсия подпрограмм; понятие активации; стековый порядок обработки данных; необходимые условия корректности рекурсивных подпрограмм; прямая и косвенная рекурсия; опережающее описание.	Индивидуальные задачи коллоквиум
6.	Структуры данных и основные алгоритмы	Комбинированные типы; множественные типы; перечислимые типы; модули; особенности использования модулей;	Индивидуальные задачи

	их обработки	преимущества использования модулей; постановка задачи сортировки; алгоритмы сортировки; информационные структуры; линейные динамические информационные структуры; моделирование линейных динамических информационных структур; пул свободной памяти; последовательное и связное распределение памяти; ссылочные типы и динамические переменные; двоичные деревья; обходы двоичных деревьев; логические и физические файлы; типизированные файлы; текстовые файлы; бестиповые файлы.	коллоквиум
7.	Объектно-ориентированное программирование	Основные понятия ООП; наследование; полиморфизм; статические методы; конструкторы; деструкторы	Индивидуальные задачи коллоквиум

2.3.2 Занятия семинарского типа – не предусмотрены.

2.3.3 Лабораторные занятия.

№	Наименование лабораторных работ	Форма текущего контроля
1	3	4
1.	Двоичное кодирование информации, двоичная и шестнадцатеричная системы счисления. Решение задач на системы счисления: преобразование чисел из одной СС в другую, арифметические операции в двоичной и шестнадцатеричной системах счисления. Метаязык Бэкуса-Наура, язык синтаксических диаграмм. Решение задач на построение описаний синтаксиса основных элементов языков программирования в БНФ и ЯСД.	Индивидуальные задачи коллоквиум
2.	Понятия: данные и их типы, константы, переменные, структура программы; операторы ввода, вывода, присваивания. Решение задач на составление линейных программ. Понятия: логический тип, условный и вариантный (оператор выбора) операторы, составной и пустой операторы. Решение задач на составление ветвящихся программ. Понятия: циклы с предусловием, с постусловием, с параметром; вложенные циклы; досрочное завершение цикла. Решение задач на составление циклических программ.	Индивидуальные задачи коллоквиум
3.	Суммы и произведения членов числовых последовательностей, последовательность Фибоначчи. Свойства делимости натуральных чисел, алгоритм Евклида. Простые и составные числа. Моделирование обработки битовой информации.	Индивидуальные задачи коллоквиум
4.	Решение задач на основные формы представления чисел. Использование стандартных подпрограмм и вычисление по формулам.	Индивидуальные задачи коллоквиум

5.	Однотипная обработка элементов массива, распознавание элементов массива с заданными свойствами, преобразование элементов массива с заданными свойствами, распознавание свойств массивов и формирование массивов с заданными свойствами. Одновременная обработка нескольких массивов.	Индивидуальные задачи коллоквиум
6.	Ввод и вывод многомерных массивов. Однотипная обработка элементов многомерного массива, решение задач на распознавание элементов многомерного массива с заданными свойствами, одновременная обработка нескольких многомерных массивов, формирование одномерных массивов на основе многомерных.	Индивидуальные задачи коллоквиум
7.	Решение задач с использованием встроенных процедур и функций для работы со строками, посимвольная обработка строковых значений, работа с массивами строк.	Индивидуальные задачи коллоквиум
8.	Понятия: подпрограмма, процедура, функция, описание и вызов подпрограммы, передача параметров. Решение задач на составление программ с подпрограммами.	Индивидуальные задачи коллоквиум
9.	Решение задач на прямую и косвенную рекурсии. Решение задач обработки структур данных, имеющих рекурсивную структуру (обработка списков, работа с деревьями, сортировки)	Индивидуальные задачи коллоквиум
10.	Решение задач упорядочения и поиска в линейных структурах данных. Оценивание алгоритмов и программ. Рекурсивные методы сортировки и поиска	Индивидуальные задачи коллоквиум
11.	Решение задач на составление программ обработки стеков, очередей, деков, списков, записей. Применение связного распределения памяти, ссылочных и динамических переменных для обработки ЛИС.	Индивидуальные задачи коллоквиум
12.	Решение задач на составление программ формирования и обработки двоичных деревьев. Решение задач с использованием процедур обхода для двоичных деревьев.	Индивидуальные задачи коллоквиум
13.	Решение задач формирования и обработки файлов.	Индивидуальные задачи коллоквиум
14.	Решение задач на применение модулей.	Индивидуальные задачи коллоквиум

2.3.4 Курсовые работы – не предусмотрены

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы

1	2	3
1	Выполнение лабораторных работ	Стандарты оформления исходного кода программ и современные интегрированные среды разработки программного обеспечения: учеб.-метод.пособие. Ю.В. Кольцов [и др.] – Краснодар: Кубанский гос.ун-т, 2017 Алексеев, Е. Программирование на Free Pascal и Lazarus : курс / Е. Алексеев, О. Чеснокова, Т. Кучер. - 2-е изд., исправ. - М. : Национальный Открытый Университет «ИНТУИТ», 2016. - 552 с. : ил. ; [Электронный ресурс]. - URL: //biblioclub.ru/index.php?page=book&id=429189

3. Образовательные технологии.

В соответствии с требованиями ФГОС программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

– Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

– Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

– Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

– Информационно-коммуникационные технологии (ИКТ) - расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

– Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

– Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

– Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

– Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

– Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

– Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

– Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

– работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

– проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

– анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

– развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
1	Л, ЛР	Занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	4
2	Л, ЛР	Занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	4
Итого			8

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации.

4.1Фонд оценочных средств для проведения текущего контроля.

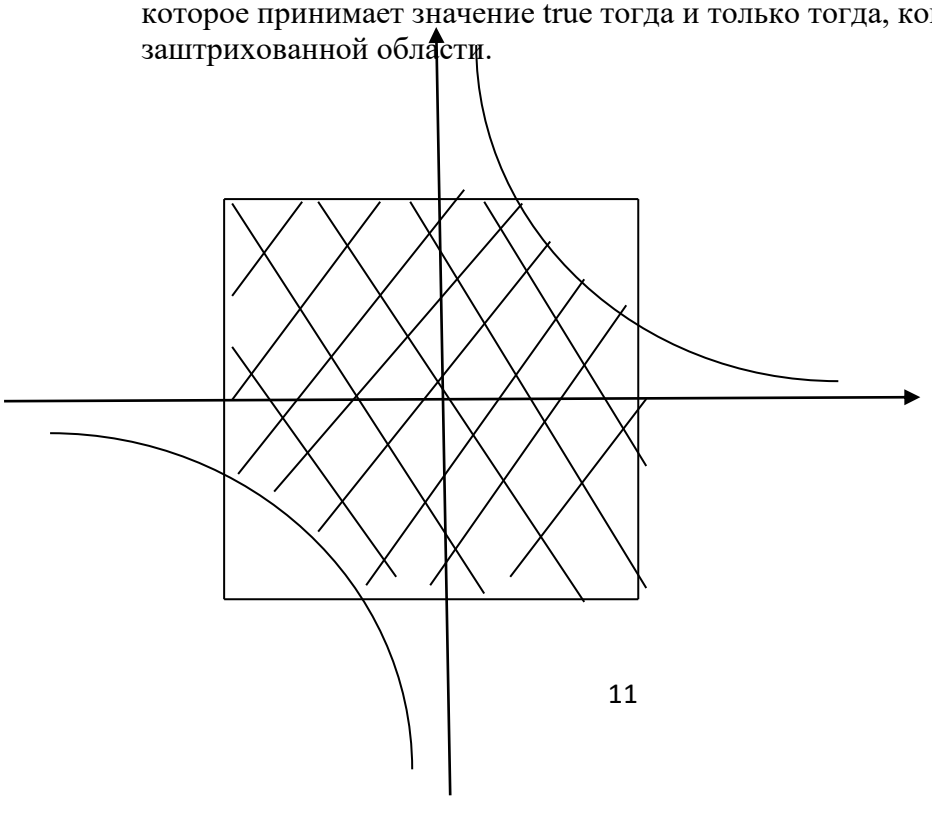
Перечень заданий текущего контроля:

Перечень компетенций, проверяемых оценочным средством:

ОПК-3 -способностью использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности

Образец коллоквиума №1

1. Понятие метаязыка. Язык синтаксических диаграмм. Примеры.
2. Оператор цикла с постусловием. Синтаксис, семантика, примеры использования.
3. Числовые типы.
4. Задача. Составить программу, которая вводит натуральное значение n и натуральные числа a_1, a_2,…,a_n и определяет сумму элементов этой последовательности, которые при делении на 7 дают остаток 1 или 2.
5. Задача. Составить программу, которая по заданной квадратной матрице mxm (где m некоторая константа) формирует одномерный массив X, каждый элемент которого равен сумме положительных элементов соответствующей строки матрицы, следующих за диагональным элементом этой строки (очевидно, что X[m]=0).
6. Пусть (x,y) – координаты точки на плоскости. Составить булевское выражение, которое принимает значение true тогда и только тогда, когда точка принадлежит заштрихованной области.



Образец коллоквиума №2.

1. Линейные динамические структуры. Стек.
2. Метод челночной сортировки.
3. Дан файл f, состоящий из целых чисел. Получить новый файл, в который входят все компоненты файла f без повторений.
4. Дан однонаправленный список целых чисел. Удалить из него все элементы, у которых четная сумма цифр.

Индивидуальные задачи по теме «Последовательности»

1. Дана последовательность из n целых чисел. Найти среднее арифметическое положительных не простых элементов последовательности, до и после которых расположены отрицательные.
2. Дана последовательность из n целых чисел. Найти произведение элементов последовательности, у которых кратная 3 сумма цифр, расположенных на четных позициях.
3. Дана последовательность целых чисел, оканчивающаяся 0. Найти произведение тех элементов, у которых произведение цифр превосходит заданное значение.
4. Дана последовательность целых чисел, оканчивающаяся 0. Найти среднее арифметическое не совершенных элементов, оканчивающихся на 3.
5. Дана последовательность из n целых чисел. Найти сумму четных элементов, которые не оканчиваются на 2, и расположены на непростых позициях.
6. Дана последовательность целых чисел, оканчивающаяся 0. Найти сумму элементов последовательности с нечетной суммой цифр, расположенных на кратных 3 позициях.
7. Дана последовательность из n целых чисел. Найти максимальный простой элемент, расположенный на нечетной позиции.
8. Дана последовательность целых чисел, оканчивающаяся 0. Найти порядковый номер минимального простого элемента.
9. Дана последовательность целых чисел, оканчивающаяся 0. Найти среднее арифметическое ее элементов, расположенных на простых позициях.
10. Дана последовательность целых чисел, оканчивающаяся 0. Найти минимальный элемент последовательности, у которого сумма цифр превосходит заданное значение.
11. Дана последовательность из n целых чисел. Найти количество элементов последовательности, у которых произведение цифр не кратно 3.
12. Дана последовательность целых чисел, оканчивающаяся 0. Вычислить произведение простых четных элементов, расположенных на кратных 3 позициях.
13. Дана последовательность целых чисел, оканчивающаяся 0. Найти количество кратных 5 элементов, расположенных на не совершенных позициях.
14. Дана последовательность целых чисел, оканчивающаяся 0. Вычислить порядковый номер максимального простого элемента.
15. Дана последовательность из n целых чисел. Найти количество элементов, расположенных на не кратных 4 позициях, сумма цифр которых кратна 5.

Индивидуальные задачи по теме «Сумма ряда»

1. Найти сумму ряда $\sum_{i=1}^{n} \frac{(-1)^{n+i}}{n}$
2. Найти сумму ряда $\sum_{i=1}^{n} \frac{(-1)^{2*i}}{i!}$
3. Найти сумму ряда $\sum_{i=1}^{n} \frac{7^{i}}{(i+5)!}$
4. Найти сумму ряда $\sum_{i=1}^{n} \frac{(-1)^{n+i}}{n}$
5. Найти сумму ряда $\sum_{i=1}^{n} \frac{(-1)^{n+i}}{(2i+3)!}$
6. Найти сумму ряда $\sum_{i=1}^{n} \frac{n^{i}+(-1)^{n+i}}{n!}$
7. Найти сумму ряда $\sum_{i=1}^{n} \frac{(x)^{i+1}}{2^{i}}$
8. Найти сумму ряда $\sum_{i=1}^{n} \frac{(3i+1)^{2}}{(2*i)!}$
9. Найти сумму ряда $\sum_{i=1}^{n} \frac{(\sin n)^{i}}{i!}$
10. Найти сумму ряда $\sum_{i=1}^{n} \frac{(-1)^{i}(\sin i)^{2+i}}{(2n-i)!}$
11. Найти сумму ряда $\sum_{i=1}^{n} \frac{(-1)^{i}*i!}{n^{i}}$
12. Найти сумму ряда $\sum_{i=1}^{n} \frac{(2)^{i-1}}{(4+2*i)!}$
13. Найти сумму ряда $\sum_{i=1}^{n} \frac{(x+2)^{2+i}}{(n+i)!}$
14. Найти сумму ряда $\sum_{i=1}^{n} \frac{(5+i)!}{n^{i}}$

Индивидуальные задачи по теме «Массивы»

1. Дан массив целых чисел, состоящий из элементов от 1 до 3. Переставить элементы массива так, чтобы сначала шли все 1, затем все 2, затем все 3. Не использовать вспомогательный массив.
2. Дан массив целых чисел. Найти наименьший элемент, индекс которого является числом Фибоначчи.
3. Дан массив целых чисел. Найти элемент, сумма цифр которого максимальна.
4. Дан массив целых чисел. Найти количество простых элементов, до и после которых располагаются числа Фибоначчи.
5. Дан массив целых чисел. Перенести в начало массива те элементы, сумма цифр которых превосходит заданное число. Не использовать вспомогательный массив.
6. Дан массив целых чисел. Если в массиве есть хотя бы одна пара дружественных чисел, то переставить эти числа в конец массива. Не использовать вспомогательный массив.
7. Дан массив целых чисел. Переставить в конец массива все простые элементы. Порядок их следования оставить прежним. Не использовать вспомогательный массив.
8. Дан массив целых чисел. Найти наибольшее простое число, индекс которого не является совершенным.
9. Дан массив целых чисел. Переставить элементы массива так, чтобы сначала шли все элементы, которые больше a_1 исходного массива, затем те, которые равны a_1 исходного массива, затем те, которые меньше. Не использовать вспомогательный массив.
10. Дан массив целых чисел. Если в результате замены отрицательных элементов исходного массива их квадратами элементы массива окажутся упорядоченными по убыванию, то получить сумму элементов исходного массива, иначе – получить их произведение.

11. Дан массив целых чисел. Переставить элементы массива так, чтобы сначала шли все отрицательные элементы, затем все неотрицательные. Порядок отрицательных и неотрицательных элементов оставить прежним. Не использовать вспомогательный массив.
12. Дан массив целых чисел. Все элементы, индексы которых являются простыми числами, заменить на значение элемента, с наименьшей суммой цифр.
13. Дан массив целых чисел. Найти индекс элемента, который имеет простое значение и максимальную сумму цифр.
14. Дан массив целых чисел. Вывести значение наибольшего элемента с наименьшей суммой цифр, из элементов, расположенных на кратных 3 позициях.
15. Дан массив целых чисел. Перенести в начало массива все совершенные элементы, а все простые элементы – в конец. Не использовать вспомогательный массив.

Индивидуальные задачи по теме «Матрицы»

1. Дана целочисленная квадратная матрица. Вывести значения всех простых элементов, сумма индексов которых является не простым числом.
2. Дана целочисленная квадратная матрица. Определить, есть ли выше побочной диагонали простое число, большее среднего арифметического всех элементов матрицы.
3. Дана целочисленная неквадратная матрица. Определить каких столбцов в матрице больше: содержащих только положительные элементы, или содержащих только отрицательные элементы?
4. Дана целочисленная квадратная матрица. Если сумма элементов выше главной диагонали больше суммы элементов выше побочной диагонали, то обнулить элементы каждой четной строки, иначе каждой нечетной.
5. Дана целочисленная неквадратная матрица. Найти произведение элементов, у которых сумма цифр не является простой.
6. Дана целочисленная неквадратная матрица. Поменять местами столбец, содержащий наибольшее значение матрицы, со столбцом, содержащим наименьшее значение матрицы. Предполагается, что наименьшее и наибольшее значения единственны.
7. Дана целочисленная квадратная матрица. Найти номера строк, элементы которых образуют симметричные последовательности.
8. Дана целочисленная неквадратная матрица. Вывести номера столбцов, элементы в которых упорядочены по возрастанию или убыванию.
9. Дано 2 целочисленных квадратных матрицы A и B. Вычислить AB-BA.
10. Дана целочисленная квадратная матрица. Заменить все элементы на побочной диагонали и выше нее значением элемента с минимальной суммой цифр.
11. Дана целочисленная квадратная матрица. Уменьшить все элементы выше главной диагонали на значение элемента с минимальной суммой цифр, а элементы ниже главной диагонали увеличить на значение элемента с максимальной суммой цифр.
12. Дана целочисленная квадратная матрица. Переставить столбцы матрицы так, чтобы они располагались по убыванию количества простых элементов в них.
13. Дана целочисленная квадратная матрица. Переставить строки матрицы так, чтобы они располагались по возрастанию количества не совершенных элементов в них.
14. Дана целочисленная квадратная матрица. Найти количество пар дружественных чисел в ней.
15. Дана целочисленная квадратная матрица. Вывести значения простых элементов вые главной диагонали, сумма индексов которых является числом Фибоначчи.

Индивидуальные задачи по теме «Функции»

1. Дана последовательность $a_1,..a_n$. Найти количество полных квадратов среди элементов последовательности.
2. Дана последовательность целых чисел $a_1,…a_n$. Найти сумму степеней пятерки в этой последовательности.
3. Дана последовательность из n целых чисел. Найти произведение простых элементов, до и после которых расположены непростые.
4. Дана последовательность из n целых чисел. Найти количество совершенных чисел, до и после которых следуют не совершенные.
5. Дана целочисленная квадратная матрица. Вывести те элементы, сумма индексов которых является числом Фибоначчи.
6. Дана целочисленная квадратная матрица. Вывести номера строк, сумма элементов в которых является симметричным числом.
7. Дана целочисленная квадратная матрица. Вывести номера строк, сумма элементов в которых не является простым числом.
8. Дана целочисленная квадратная матрица. Найти произведение тех элементов, которые являются числами Фибоначчи.
9. Дана целочисленная квадратная матрица. Найти количество полных квадратов среди ее элементов.
10. Дан массив целых чисел. Найти количество симметричных элементов в нем.
11. Дана целочисленная квадратная матрица. Найти количество элементов, которые являются степенями 3.
12. Дана последовательность $a_1,..a_n$. Найти количество элементов последовательности, в записи которых встречается ровно три восьмерки.
13. Дан массив целых чисел. Найти количество пар дружественных чисел в нем.
14. Дана целочисленная квадратная матрица. Найти произведение простых элементов с нечетной суммой цифр.
15. Дан массив целых чисел. Найти произведение тех элементов, которые являются полными квадратами простых чисел.

Индивидуальные задачи по теме «Однонаправленные списки»

1. Дано два однонаправленных списка целых чисел. Сформировать третий список, содержащий простые числа исходных списков.
2. Дано два однонаправленных списка целых чисел. Сформировать третий список, содержащий совершенные числа исходных списков.
3. Дано два однонаправленных списка целых чисел. Сформировать третий список, содержащий числа Фибоначчи исходных списков.
4. Дано два однонаправленных списка целых чисел. Сформировать третий список, содержащий пары дружественных чисел, одно число пары находится в первом исходном списке, второе - во втором исходном списке.
5. Дано два однонаправленных списка целых чисел. В первый список после максимального элемента вставить второй список, элементы которого расположены в обратном порядке.
6. Дано два однонаправленных списка целых чисел. Из первого списка удалить все
   элементы, большие максимального элемента второго списка и меньшие минимального элемента второго списка.
7. Дано два однонаправленных списка целых чисел. Удалить во втором списке все элементы, большие среднего арифметического положительных элементов первого списка и продублировать все элементы второго списка, меньшие среднего арифметического четных элементов первого списка.
8. Дано два однонаправленных списка целых чисел. Заменить все элементы с четной суммой цифр из первого списка на наибольший элемент второго списка.

9. Дано два однонаправленных списка целых чисел. Увеличить все элементы с четным количеством цифр из первого списка на значение наименьшего элемента
   второго списка.
10. Дан однонаправленный список целых чисел. Отсортировать его по возрастанию.
11. Описать метод, который
    а) вставляет в список L новый элемент E1 за каждым вхождением элемента E.
    б) удаляет из списка L все отрицательные элементы.
12. Описать метод, который
    а) вставляет в непустой список L пару новых элементов E1 и E2 перед его последним элементом
    б) удаляет из списка L за каждым вхождением элемента E один элемент, если такой есть и он отличен от E.
13. Описать метод, который
    а) вставляет в непустой список L, элементы которого упорядочены по не убыванию новый элемент E так, чтобы сохранилась упорядоченность.
    б) удаляет из списка L первый отрицательный элемент.
14. Описать метод, который
    а) проверяет, есть ли в списке L хотя бы два одинаковых элемента.
    б) добавляет в конец списка L1 все элементы списка L2.
15. Описать метод, который
    а) переносит в начало непустого списка L его первый элемент.
    б) вставляет в список L за первым вхождением элемента E все элементы списка L1,если E входит в L.

Индивидуальные задачи по теме «Двунаправленные списки»

1. Дан двунаправленный кольцевой список целых чисел. Вставить перед каждым элементом, являющимся простым числом, элемент равный количеству цифр этого простого числа.
2. Дан двунаправленный список целых чисел. Переставить элементы списка так, чтобы сначала следовали элементы со значением меньше среднего арифметического, а затем элементы со значением больше.
3. Дан двунаправленный список целых чисел. Переставить элементы списка так, чтобы сначала следовали элементы с простыми значениями, затем элементы с четными, а затем с нечетными. Остальные элементы списка удалить.
4. Дан двунаправленный список целых чисел. Переставить элементы списка так, чтобы сначала следовали элементы являющиеся числами Фибоначчи, а затем остальные элементы списка.
5. Дан двунаправленный список целых чисел. Получить однонаправленный список, содержащий все элементы исходного с кратным 3 количеством цифр, порядок необходимо изменить на обратный.
6. Дан двунаправленный список целых чисел. Получить однонаправленный список, содержащий все элементы исходного, порядковые номера которых являются простыми числами, а значения не являются простыми.
7. Дан двунаправленный список целых чисел. Отсортировать элементы списка в порядке убывания.
8. Дан двунаправленный список целых чисел. Получить однонаправленный список, содержащий все элементы исходного, значения которых не превышают среднего арифметического положительных элементов исходного списка.

9. Дан двунаправленный список целых чисел. Получить однонаправленный список, содержащий все элементы исходного, значения которых отличаются от разности (max-min) не более чем на 3.
10. Дан кольцевой двунаправленный список.
    а) Из списка L, содержащего не менее двух элементов, удалить все элементы, у которых одинаковые соседи (первый и последний элемент считать соседями).
    б) в списке L удвоить каждое вхождение элемента E.
11. Дан кольцевой двунаправленный список.
    а) построить список L по однонаправленному списку L1.
    б) в конец непустого списка L добавить все его элементы, располагая их в обратном порядке (например, L=l 2 3, ответ=1 2 3 3 2 1).

Индивидуальные задачи по теме «Двунаправленные деревья»

1. Дано дерево двоичного поиска. Найти количество совершенных листьев дерева, больших разности максимального и минимального элемента.
2. Дано дерево двоичного поиска. Если в дереве встречается хотя бы два кратных 3 элемента, то увеличить простые элементы дерева вдвое.
3. Дано дерево двоичного поиска. Найти количество элементов дерева меньших среднего арифметического листьев дерева.
4. Дано дерево двоичного поиска. Если все элементы дерева не кратны 4, то заменить все элементы с четной суммой цифр их квадратами.
5. Дано дерево двоичного поиска. Если все элементы дерева кратны 7, то заменить все отрицательные элементы их модулями.
6. Дано дерево двоичного поиска. Найти количество четных элементов дерева, принадлежащих [a; b] и количество нечетных листьев не принадлежащих [a,b].
7. Дано дерево двоичного поиска. Если в дереве встречается хотя бы четыре непростых элемента, то уменьшить нечетные элементы дерева вдвое.
8. Дано дерево двоичного поиска. Верно ли, что сумма элементов дерева больших среднего арифметического листьев дерева равно заданному числу N.
9. Дано дерево двоичного поиска. Если все элементы дерева кратны 3, то заменить все элементы с нечетным количеством цифр их квадратами.
10. Дано дерево двоичного поиска. Если все листья дерева кратны 5, то заменить все отрицательные элементы их модулями.
11. Дано дерево двоичного поиска. Найти количество четных листьев дерева, принадлежащих [a; b] и количество четных элементов не принадлежащих [a,b].
12. Дано дерево двоичного поиска. Если в дереве встречается не более трех чисел Фибоначчи, то уменьшить нечетные листья дерева втрое.
13. Дано дерево двоичного поиска. Описать функцию, которая подсчитывает число вершин на N - ом уровне непустого дерева. (Корень - вершина 0 - го уровня).
14. Дано дерево двоичного поиска. Если в дереве встречается не более 2 совершенных элементов, то уменьшить кратные 3 листья дерева втрое.
15. Дано дерево двоичного поиска. Найти произведение листьев дерева с нечетной суммой цифр, больших удвоенного минимального элемента.

Индивидуальные задачи по теме «Объектно-ориентированное программирование»

1. Создать объектный тип студент, имеющий имя, курс и идентификационный номер. Определить конструкторы, деструктор и функцию вывода на экран. Создать производный объектный тип - студент-дипломник, имеющий тему диплома. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию

вывода на экран. Определить функции переназначения названия диплома и идентификационного номера.

2. Создать объектный тип: время с полями: час (0-23), минуты (0-59), секунды (0-59). Объектный тип имеет конструктор. Функции установки времени, функции получения часа, минуты и секунды, а также две функции вывода на экран: вывода по шаблону: часов 18 минут 3 секунды и 4 p.m. 18 минут 3 секунды. Функции установки полей объектного типа должны проверять корректность задаваемых параметров. Функция дает приращение на 1 сек. 1 мин, и 1 час.

3. Создать объектный тип колесо, имеющий радиус. Определить конструкторы и метод доступа. Создать объектный тип машин, содержащий объектный тип колесо. Дополнительно есть марка, цена. Определить конструкторы и деструктор. Определить производный объектный тип грузовик, имеющий дополнительно грузоподъемность. Определить конструкторы, деструкторы и функцию вывода на экран.

4. Создать объектный тип точка, имеющая координаты. Определить конструкторы, деструктор и функцию вывода на экран. Создать производный объектный тип – цветная точка, имеющий цвет точки. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию вывода на экран. Определить функции переназначения цвета и координат точки, вывода точки на экран.

5. Создать объектный тип окно, имеющий координаты верхнего левого и нижнего правого угла, цвет фона. Определить конструкторы, деструктор и функцию вывода на экран. Создать производный объектный тип - окно с меню, имеющий строку меню. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию вывода на экран. Определить функции переназначения цвета фона и строки меню.

6. Создать объектный тип жесткий диск, имеющий объем (Мбайт). Определить конструкторы и метод доступа. Создать объектный тип компьютер, содержащий объектный тип жесткий диск. Дополнительно есть марка, цена. Определить конструкторы и деструктор. Определить производный объектный тип компьютеров с монитором, имеющий дополнительно размер монитора. Определит конструкторы, деструкторы и функцию вывода на экран.

7. Создать объектный тип жидкость, имеющий название, плотность. Определить конструкторы, деструктор и функцию вывода на экран. Создать производный объектный тип - спиртные напитки, имеющий крепость. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию вывода на экран. Определить функции переназначения плотности и крепости.

8. Создать объектный тип - дата с полями: день (1-31), месяц (1-12), год (целое число). Объектный тип имеет конструктор. Функции установки дня, месяца и года. Функции получения дня, месяца и года, а также две функции вывода на экран: вывода по шаблону: 5 января 1997 года и 05.01.1997. Функции установки полей объектного типа должны проверять корректность задаваемых параметров. Функция дает приращение на 1 день.

9. Создать колоду карт. Конструкторы колоды должны инициализировать колоду упорядочено и случайным образом. Создать производный объектный тип от колоды - пасьянс, в котором выбираются по три карты и, если две крайние одного цвета, то их выбрасывают. Всю колоду проходят три раза.

10. Создать объектные типы четырехугольников, квадратов и прямоугольников. Создать из них иерархию. Определить функции вывода на экран, конструкторы и деструкторы, вычисление площади и периметра.

4.2 Фонд оценочных средств для проведения промежуточной аттестации.

Зачетно-экзаменационные материалы для промежуточной аттестации (зачет, экзамен)

Перечень компетенций, проверяемых оценочным средством:

ОПК-3 -способностью использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности

Образец контрольной работы №1

1. Дана последовательность целых чисел, оканчивающаяся 0. Найти сумму четных элементов, до и после которых расположены элементы, не оканчивающиеся на 3.
2. Дана последовательность из n целых чисел. Найти среднее арифметическое элементов на не простых позициях.
3. Дано натуральное n и действительное x. Вычислить $\sum_{i=1}^{n} \frac{-1^{2i}x}{sin^{i}x}$
4. Дан массив целых чисел. Заменить все элементы, оканчивающиеся на 3 на значение наибольшего простого элемента.
5. Дан массив целых чисел. Если он симметричный, то вычислить количество простых элементов в нем, иначе – сумму элементов с кратной 3 суммой цифр.
6. Дана целочисленная квадратная матрица. Получить одномерный логический массив, элемент которого равен true, если в соответствующей строке матрицы ровно 2 простых элемента, расположенных после диагонального элемента, и false иначе.
7. Решить с использованием функции: дана целочисленная квадратная матрица. Найти произведение элементов, расположенных выше побочной диагонали, которые являются числами Фибоначчи.

Образец контрольной работы №2

1. Дан двунаправленный кольцевой список. Между равными элементами вставить новый элемент, равный их сумме.
2. Дан однонаправленный список целых чисел. Рекурсивно найти количество кратных 3 элементов.
3. Дано дерево двоичного поиска. Если в дереве ровно 5 четных листьев, то заменить все отрицательные элементы их модулем, иначе увеличить все нечетные элементы вдвое.
4. Дан объект строка, методы – ввод, вывод, удаление всех точек из строки, вставка после каждого пробела заданного символа.

Перечень вопросов для подготовки к экзамену

1. Информация.
2. Методы получения информации.
3. Свойства информации.
4. Количество информации.
5. Передача и обработка информации.
6. Информационные системы.
7. Алгоритмы. Свойства алгоритмов.
8. Трансляция алгоритмов.
9. Языки программирования. Их свойства.

10. Основные аспекты языков программирования.
11. Метаязыки. Метаязык Бекуса-Наура.
12. Метаязыки. Язык синтаксических диаграмм.
13. Структура простейшей программы.
14. Типы данных языка C++, их классификация.
15. Краткая характеристика конструируемых типов.
16. Определение переменных и констант.
17. Целочисленные типы.
18. Вещественные типы.
19. Основные встроенные арифметические функции.
20. Логический тип.
21. Классификация операторов языка C++. Простые операторы.
22. Условный оператор.
23. Оператор цикла с предусловием.
24. Оператор цикла с постусловием.
25. Оператор цикла с параметром.
26. Символьный тип.
27. Оператор выбора.
28. Конструктор типа.
29. Интервальный тип.
30. Регулярный тип.
31. Двумерные массивы.
32. Строковый тип.
33. Встроенные подпрограммы для работы со строками.
34. Подпрограммы.
35. Процедуры. Формальные и фактические параметры.
36. Параметры значения и параметры переменные.
37. Функции.
38. Рекурсия.
39. Прямая и косвенная рекурсия.
40. Комбинированные типы.
41. Перечислимые типы.
42. Рекурсия подпрограмм.
43. Прямая и косвенная рекурсия
44. Использование имен подпрограмм в качестве параметров.
45. Комбинированные типы.
46. Перечислимые типы.
47. Множественные типы.
48. Реализация множеств
49. Ввод-вывод информации. Система управления вводом-выводом.
50. Методы доступа.
51. Метки файлов.
52. Открытие и закрытие файлов.
53. Буферизация ввода-вывода
54. Логические и физические файлы.
55. Общие операции над файлами
56. Краткая характеристика основных классов файлов языка C++.
57. Типизированные файлы.
58. Бестиповые файлы.
59. Файлы прямого доступа.
60. Текстовые файлы.
61. Понятие информационной структуры.

62. Линейные динамические информационные структуры: Стек, Очередь, Дек.
63. Моделирование ЛДИС средствами языка С++. Моделирование стека
64. Моделирование очереди.
65. Кольцевой буфер.
66. Реализация двух однотипных стеков.
67. Понятие пула свободной памяти.
68. Связное распределение памяти. Моделирование СРП в С++.
69. Преимущества и недостатки связного распределения памяти.
70. Моделирование основных операций над ПСП.
71. Реализация m стеков с общим пулом свободной памяти.
72. Фрагментация памяти.
73. Ссылочные типы. Основные понятия и операции. Динамические переменные.
74. Процедуры New и Dispose. Понятие мусора.
75. Представление и обработка динамических списков в Паскале.
76. Двоичные деревья.
77. Деревья двоичного поиска.
78. Формирование дерева двоичного поиска.
79. Алгоритмы сортировки. Метод линейного поиска с обменом.
80. Алгоритмы сортировки. Метод челночной сортировки.
81. Алгоритмы сортировки. Метод парных сравнений.
82. Алгоритмы сортировки. Сортировка Хоара.
83. ООП. Основные понятия.
84. Инкапсуляция объектов.
85. Наследование объектов.
86. Полиморфизм объектов.
87. Статические методы.
88. Виртуальные методы.
89. Конструкторы. Деструкторы.

Перечень задач для подготовки к экзамену

1. Дана последовательность из n целых чисел. Найти среднее арифметическое четных элементов на кратных 3 и не кратных 5 позициях.
2. Дан массив целых чисел. Заменить все отрицательные элементы на значение максимального элемента массива.
3. Дан массив целых чисел. Заменить все элементы на четных позициях на заданное значение.
4. Дана целочисленная квадратная матрица. Заменить все отрицательные элементы выше главной диагонали на заданное значение.
5. Дано 2 целочисленных квадратных матрицы. Увеличить элементы строки на заданное значение, если сумма нечетных элементов строки на четных позициях первой матрицы не превосходит первого элемента соответствующей строки второй матрицы. (Подпрограмма)
6. Дано 2 целочисленных квадратных матрицы. Уменьшить элементы строки первой матрицы вдвое, если сумма нечетных элементов строки первой матрицы не менее последнего элемента соответствующей строки второй матрицы. (Подпрограмма)
7. Дано 2 целочисленных квадратных матрицы. Увеличить элементы строки первой матрицы на заданное значение, если среднее арифметическое положительных

элементов строки первой матрицы не превосходит последнего элемента соответствующей строки второй матрицы. (Подпрограмма)

8. Дан массив строк. Вычислить количество не симметричных срок, вторая половина которых не содержит цифр. (Подпрограмма)
9. Дан массив строк. Напечатать номера строк, первая половина которых является симметричной, и длина которых является простым числом. (Подпрограмма)
10. Дан массив строк. Вычислить количество строк, не содержащих букв в первой половине, длина которых является не совершенным числом. (Подпрограмма)
11. Дана целочисленная квадратная матрица. Получить одномерный логический массив, элемент которого равен true, если соответствующий столбец матрицы содержит не менее 3-х совершенных элементов, и false иначе.
12. Дана целочисленная квадратная матрица. Получить одномерный логический массив, элемент которого равен true, если соответствующая строка матрицы содержит не более 5-ти простых элементов, и false иначе.
13. Дана целочисленная квадратная матрица. Получить одномерный массив, элемент которого равен количеству не простых элементов соответствующей строки матрицы, расположенных после диагонального элемента.
14. Дана целочисленная квадратная матрица. Получить одномерный массив, элемент которого равен произведению не совершенных элементов соответствующего столбца матрицы, расположенных ниже диагонального элемента.
15. Дан массив строк. Если строка содержит не более 3-х пробелов, то заменить все точки на пробелы, иначе вычислить количество цифр, после которых идут *.
16. Дан массив строк. Если в строке содержится заданное число точек, то заменить все точки на *, иначе вставить после каждой точки еще две точки.
17. Дан массив строк. Если строка симметрична, то заменить все буквы на заданный символ, иначе вычислить количество точек, после которых расположены пробелы.
18. Дан массив строк. Если строка не симметрична, то заменить все запятые после которых идут пробелы на заданный символ, иначе вставить перед каждой цифрой пробел.
19. В непустом двоичном дереве найти количество узлов, не являющихся листьями.
20. Найти разность между максимальным и минимальным элементом в дереве двоичного поиска.

Компонентом промежуточной аттестации по дисциплине «Программирование» является контрольная работа. Максимальное количество баллов, которые студент может получить за правильное решение одного задания контрольной работы, составляет 2 балла. Общее количество баллов за контрольную работу вычисляется суммированием баллов за каждое задание.

Описание	Баллы
Предоставлен работоспособный программный код, студент может пояснить ход решения, знает назначение команд, может изменить некоторые условия по просьбе преподавателя.	2
Программный код может быть не работоспособен, однако алгоритм решения задачи корректный, студент может пояснить ход решения, знает назначение некоторых команд	1
Программный код не работает, алгоритм решения не верный, студент не знает назначения отдельных команд	0

Критерии оценки:

- **оценка «зачтено»**: студент получил не менее 4 баллов за коллоквиум и не менее 5 баллов индивидуальные задачи, а также не менее 6 баллов за контрольную работу.

- **оценка «незачет»**: студент получил менее 4 баллов за коллоквиум или менее 5 баллов за индивидуальные задачи, или менее 6 баллов за контрольную работу.

Оценка	
Незачет	**Зачтено**
• студент получил менее 4 баллов за коллоквиум • менее 5 баллов за индивидуальные задачи • менее 6 баллов за контрольную работу	• студент получил не менее 4 баллов за коллоквиум • не менее 5 баллов за индивидуальные задачи • не менее 6 баллов за контрольную работу

Методические рекомендации определяющие процедуры оценивания на экзамене

Итоговой формой контроля сформированности компетенций у обучающихся по дисциплине является экзамен. Студенты обязаны сдать экзамен в соответствии с расписанием и учебным планом

ФОС промежуточной аттестации состоит из контрольной работы к зачету и вопросов и задач к экзамену по дисциплине.

Экзамен по дисциплине преследует цель оценить работу студента за курс, получение теоретических знаний, их прочность, развитие творческого мышления, приобретение навыков самостоятельной работы, умение применять полученные знания для решения практических задач.

Форма проведения экзамена: письменно.

Экзаменатору предоставляется право задавать студентам дополнительные вопросы по всей учебной программе дисциплины.

Результат сдачи экзамена заноситься преподавателем в экзаменационную ведомость и зачетную книжку.

Методические рекомендации определяющие процедуры оценивания задачи на экзамене

Компонентом промежуточного контроля по дисциплине «Программирование» является задача в виде программного кода. Максимальное количество баллов, которые студент может получить за правильное решение задачи, составляет 2 балла.

Описание	Баллы
Предоставлен работоспособный программный код, студент может пояснить ход решения, знает назначение команд, может изменить некоторые условия по просьбе преподавателя.	2
Программный код может быть не работоспособен, однако алгоритм решения задачи корректный, студент может пояснить ход решения, знает назначение некоторых команд	1
Программный код не работает, алгоритм решения не верный, студент не знает назначения отдельных команд	0

Критерии оценки:

Оценка			
Неудовлетворительно	**Удовлетворительно**	**Хорошо**	**Отлично**
• Не выполнена задача и не ответил ни на один вопрос билета • непонимание сущности излагаемых вопросов, грубые ошибки в ответе, неуверенные и неточные ответы на дополнительные вопросы экзаменаторов.	• Задача выполнена на 1 балл • частично ответил на два вопроса билета • отвечает на дополнительные вопросы кратко, допуская неточности	• достаточно полно ответил на два вопроса билета • задача выполнена на 1-2 балла • твёрдые и достаточно полные ответы на дополнительные вопросы	• исчерпывающий ответ на 2 вопроса билета, с примерами и пояснениями • задача решена на 2 балла • исчерпывающие, логически последовательные, полные, правильные и конкретные ответы на дополнительные вопросы;

Перечень компетенций, проверяемых оценочным средством:

ОПК-3 - способностью использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины.

5.1 Основная литература:

1. Программирование и основы алгоритмизации : учебное пособие / В.К. Зольников, П.Р. Машевич, В.И. Анциферова, Н.Н. Литвинов ; Министерство образования и науки Российской Федерации, Федеральное агентство по образованию, Государственное образовательное учреждение высшего профессионального образования «Воронежская государственная лесотехническая академия». - Воронеж : Воронежская государственная лесотехническая академия, 2011. - 341 с. : ил. ; То же - URL: http://biblioclub.ru/index.php?page=book&id=142309
2. Забуга, А.А. Теоретические основы информатики / А.А. Забуга. - Новосибирск : НГТУ, 2013. - 168 с. - ISBN 978-5-7782-2312-7 ; То же - URL: http://biblioclub.ru/index.php?page=book&id=258592.
3. Седжвик, Р. Алгоритмы на C++ / Р. Седжвик. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 1773 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429164
4. Синицын, С.В. Основы разработки программного обеспечения на примере языка С / С.В. Синицын, О.И. Хлытчиев. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 212 с. : схем., ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=429186
5. Белоцерковская, И.Е. Алгоритмизация. Введение в язык программирования С++ / И.Е. Белоцерковская, Н.В. Галина, Л.Ю. Катаева. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 197 с. : ил. ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=428935
6. Лубашева, Т.В. Основы алгоритмизации и программирования : учебное пособие / Т.В. Лубашева, Б.А. Железко. - Минск : РИПО, 2016. - 378 с. : ил. - Библиогр. в кн. - ISBN 978-985-503-625-9 ; То же [Электронный ресурс]. - URL: http://biblioclub.ru/index.php?page=book&id=463632

5.2.Дополнительная литература:

1. Таланов, А.В. Графы и алгоритмы / А.В. Таланов, В.Е. Алексеев. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 154 с. : ил. - Библиогр. в кн. - ISBN 5-9556-0066-3 ; То же - URL: http://biblioclub.ru/index.php?page=book&id=428827

2. Мейер, Б. Инструменты, алгоритмы и структуры данных / Б. Мейер. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 543 с. : схем., ил. - Библиогр. в кн. ; То же - URL: http://biblioclub.ru/index.php?page=book&id=429033

3. Теория алгоритмов : лабораторный практикум / Министерство образования и науки РФ, Федеральное государственное автономное образовательное учреждение высшего образования «Северо-Кавказский федеральный университет» ; сост. А.А. Брыкалова. - Ставрополь : СКФУ, 2016. - 134 с. - Библиогр. в кн. ; То же - URL: http://biblioclub.ru/index.php?page=book&id=467401

4.Информатика и программирование : учебное пособие / Р.Ю. Царев, А.Н. Пупков, В.В. Самарин, Е.В. Мыльникова ; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск : Сибирский федеральный университет, 2014. - 132 с. : ил., табл., схем. - Библиогр. в кн.. - ISBN 978-5-7638-3008-8 ; То же - URL: http://biblioclub.ru/index.php?page=book&id=364538

5.Информатика : учебное пособие / Министерство образования и науки Российской Федерации, Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Тамбовский государственный технический

университет». - Тамбов : Издательство ФГБОУ ВПО «ТГТУ», 2015. - 159 с. : ил. - Библ. в кн. - ISBN 978-5-8265-1490-0 ; То же -
URL: http://biblioclub.ru/index.php?page=book&id=445045

5.3. Периодические издания:

1. Прикладная информатика
2. Проблемы передачи информации
3. Программные продукты и системы
4. Программирование
5. COMPUTATIONAL NANOTECHNOLOGY (ВЫЧИСЛИТЕЛЬНЫЕ НАНОТЕХНОЛОГИИ)
6. COMPUTERWORLD РОССИЯ
7. WINDOWS IT PRO / RE

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины.

1. Самоучитель Pascal — URL: http://www.raaar.ru/unix/pascal/samouch_pascal.html
2. Программирование на языке Pascal – URL: http://intuit.valrkl.ru/course-112/index.html

7. Методические указания для обучающихся по освоению дисциплины.

По курсу предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал. После прослушивания лекции рекомендуется самостоятельно выполнить на компьютере программные примеры, приводимые в лекции.

Важнейшим этапом курса является самостоятельная работа студентов. При самостоятельной работе необходимо изучить литературу, приведенную в перечнях выше, дляосмысления вводимых понятий, анализа предложенных подходов и методов разработки программ.

Разрабатывая решение новой задачи, студент должен уметь выбрать эффективные и надежные структуры данных для представления информации, подобрать соответствующие алгоритмы для их обработки, учесть специфику языка программирования, на котором будет выполнена реализация. Студент должен уметь выполнять тестирование и отладку алгоритмов решения задач с целью обнаружения, и устранения в них ошибок.

На лабораторных занятиях в качестве систем программирования для решения задач и изучения методов и алгоритмов, приведенных в лекциях, рекомендуется использовать среды разработки FreePascal или ABCPascal. Для эффективного программирования рекомендуется использовать встроенные отладчики.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.

8.1 Перечень информационных технологий.

– Использование электронных презентаций при проведении лекционных занятий.

8.2 Перечень необходимого программного обеспечения.

– Интегрированная среда разработки программ (FreePascal, ABCPascal).

– Программное обеспечение для безопасного отображения презентаций

8.3Перечень информационных справочных систем:

1. Электронная библиотечная система eLIBRARY.RU (http://www.elibrary.ru)/

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.

№	Вид работ	Материально-техническое обеспечение дисциплины и оснащенность
1.	Лекционные занятия	Лекционная аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) и соответствующим программным обеспечением
2.	Лабораторные занятия	Аудитория, оснащенная персональными компьютерами с установленным программным обеспечением и безопасным выходом в Интернет.
3.	Групповые (индивидуальные) консультации	Аудитория, оснащенная персональными компьютерами с установленным программным обеспечением
4.	Текущий контроль, промежуточная аттестация	Аудитория, оснащенная персональными компьютерами с установленным программным обеспечением
5.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.