

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

УТВЕРЖДАЮ  
Проректор по учебной работе, качеству  
образования и первым проректор  
Магуров Г.А.  
подпись  
«27» апреля 2018 г.



## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

### **Б1.Б.05 ОСНОВЫ ПРОГРАММИРОВАНИЯ**

Направление подготовки/специальность 02.03.02 «Фундаментальная информатика и информационные технологии»

Направленность (профиль) / специализация «Вычислительные технологии»

Программа подготовки академическая

Форма обучения очная

Квалификация (степень) выпускника бакалавр

Краснодар 2018

Рабочая программа дисциплины Основы программирования составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.02 «Фундаментальная информатика и информационные технологии»

Программу составил(и):

доцент кафедры информационных технологий КубГУ, к.п.н., доцент



Добровольская Н.Ю.

Рабочая программа дисциплины Основы программирования 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 вычислительных технологий, протокол № 7 «03» апреля 2018г.

Заведующий кафедрой (выпускающей)



Миков А.И.

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 1 «20» апреля 2018г.

Председатель УМК факультета



Малыхин К.В.

Рецензенты:

доцент кафедры математического моделирования ФГБОУ «КубГУ»

Рубцов С. Е.

заведующий кафедрой СГЕНД СКФ ФГБОУ ВО «Российский государственный университет правосудия»

Бегларян М. Е.

## **1 Цели и задачи изучения дисциплины.**

### **1.1 Цель освоения дисциплины.**

Формирование информационной картины мира, основанной на понимании сущности и значения информации в развитии современного информационного общества; приобретение устойчивых навыков сбора, хранения и обработки информации.

Создание необходимой основы для использования современных средств вычислительной техники и прикладных программ при изучении студентами естественнонаучных, общепрофессиональных и специальных дисциплин

Освоение, предусмотренного программой, теоретического материала и приобретение практических навыков использования информационных систем и технологий на базе современных ПК

### **1.2 Задачи дисциплины.**

Основные задачи курса на основе системного подхода:

- систематическое изучение языков программирования высокого уровня;
- формирование у студентов знаний, умений и владений в области алгоритмизации задач вычислительного характера и задач автоматизированной обработки данных;
- изучение сложных структур данных и их применение для решения различных задач обработки данных на ЭВМ;
- расширение представлений о современном программном обеспечении, языках программирования высокого уровня;
- знакомство с современными технологиями программирования – ООП технологией и визуальным конструированием программ.

Цели и задачи данного курса вытекают из необходимости практического применения ЭВМ и закрепления полученных умений и навыков работы со средствами вычислительной техники, применения различных языков и методов программирования для исследования математических и информационных моделей.

### **1.3 Место дисциплины в структуре образовательной программы.**

Дисциплина «Основы программирования» относится к базовой части Блока1 "Дисциплины (модули)" учебного плана.

Дисциплина «Основы программирования» является логически и содержательно-методически связана с такими дисциплинами как «Дискретная математика», «Конструирование алгоритмов и структур данных», «Парадигмы программирования». Данная дисциплина позволяет заложить основу для изучения программистских дисциплин профессионального цикла, предшествует таким дисциплинам как «Конструирование алгоритмов и структур данных», «Парадигмы программирования». Является логически связанной с математическими дисциплинами, рассматривает объекты таких дисциплин как «Дискретная математика» с точки зрения программирования.

Для успешного освоения дисциплины студент должен иметь базовую подготовку по информатике в объеме программы средней школы.

Обучающийся должен:

- уметь представлять число в различных системах счисления и выполнять арифметические действия в них;
- уметь строить элементарные линейные алгоритмы и блок-схемы алгоритмов;
- уметь кодировать информацию;
- уметь решать логические задачи.

### **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)

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
1.	ОПК-2	Способностью применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий	1) способы применения в профессиональной деятельности современных языков программирования и языков баз данных, методологий системной инженерии 2) назначение и состав программного обеспечения персональных компьютеров; основные этапы решения задач на ПК; 3) современные профессиональные стандарты информационных технологий; 4) современные интегрированные среды для решения основных классов инженерных и экономических задач; 5) основные приемы алгоритмизации и программирования на языках	6) управлять персональным компьютером в автономном режиме и в составе компьютерной сети; создавать и редактировать текстовые документы, электронные таблицы и базы данных с помощью одного из текстовых редакторов; 7) пользоваться сетевыми технологиями, библиотеками и пакетами программ; 8) подготовить задачу для решения на ПК, включая ее математическую постановку, выбор метода решения, описание алгоритма и составление программы; 9) самостоятельно применять компьютеры для решения учебных задач, используя для этого	10) принципами конструирования систем автоматизации проектирования, электронных библиотек и коллекций, сетевых технологий, 11) навыками работы пользователя и программиста в интегрированных средах, использующих "оконный интерфейс"; 12) основами использования компьютерной техники и информационными технологиями.

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
			высокого уровня; возможности, принципы построения и правила использования наиболее распространенных пакетов прикладных программ общего назначения	соответствующие инструментальные средства; использовать возможности современной вычислительной техники и программного обеспечения для решения инженерно-технических задач и задач производственной и управленческой деятельности	

## 2. Структура и содержание дисциплины.

### 2.1 Распределение трудоёмкости дисциплины по видам работ.

Общая трудоёмкость дисциплины составляет 10 зач.ед. (360 часов), их распределение по видам работ представлено в таблице (для студентов ОФО).

Вид учебной работы	Всего часов	Семестры (часы)			
		1	2		
<b>Контактная работа, в том числе:</b>					
<b>Аудиторные занятия (всего):</b>	<b>174</b>	<b>72</b>	<b>102</b>		
Занятия лекционного типа	140	72	68	-	-
Лабораторные занятия	34	-	34	-	-
Занятия семинарского типа (семинары, практические занятия)	-	-	-	-	-
	-	-	-	-	-
<b>Иная контактная работа:</b>					
Контроль самостоятельной работы (КСР)	12	8	4		
Промежуточная аттестация (ИКР)	0,8	0,3	0,5		
<b>Самостоятельная работа, в том числе:</b>					
Курсовая работа	-	-	-	-	-
Проработка учебного (теоретического) материала	34	24	10	-	-
Выполнение индивидуальных заданий (индивидуальные задачи, коллоквиум, контрольные работы)	50	30	20	-	-
Реферат	-	-	-	-	-
Подготовка к текущему контролю	17,8	10	7,8	-	-

<b>Контроль:</b>						
Подготовка к экзамену		71,4	35,7	35,7		
<b>Общая трудоемкость</b>	<b>час.</b>	<b>360</b>	<b>180</b>	<b>180</b>	<b>-</b>	<b>-</b>
	<b>в том числе контактная работа</b>	<b>186,8</b>	<b>80,3</b>	<b>106,5</b>		
	<b>зач. ед</b>	<b>10</b>	<b>5</b>	<b>5</b>		

## 2.2 Структура дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы дисциплины, изучаемые в 1 семестре (*очная форма*)

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа		Внеаудиторная работа	
			Л	ЛР	СРС	контроль
1	2	3	4	5	6	7
1.	Основные понятия языков программирования	50	22		18	10
2.	Алгоритмические основы информатики. Алгоритмы, их свойства и средства описания. Визуализация алгоритмов. Базовые алгоритмические структуры	54	26		18	10
3.	Основные структуры данных	50	24		16	10
4.	Обзор изученного материала	17,7			12	5,7
5.	Контроль самостоятельной работы (КСР)	8				
6.	Промежуточная аттестация (ИКР)	0,3				
	<i>Итого по дисциплине:</i>	180	72		64	35,7

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

Разделы дисциплины, изучаемые в 2 семестре (*очная форма*)

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа		Внеаудиторная работа	
			Л	ЛР	СРС	контроль
1	2	3	4	6	7	7
1	Средства объектно - ориентированного программирования	50	20	10	10	10
2	Линейные динамические структуры	56	24	12	10	10
3	Структуры данных. Алгоритмы на графах.	57	24	10	10	10
4	Обзор изученного материала и прием зачета	14,5		2	7,8	5,7
5.	Контроль самостоятельной работы (КСР)	4				
6.	Промежуточная аттестация (ИКР)	0,5				
	<i>Итого по дисциплине:</i>	180	68	34	37,8	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	Основные понятия языков программирования	Элементы и структуры данных, алфавит, имена, выражения, операции, операторы. Структуры программ. Аппарат подпрограмм. Реализация логических структур.	К
2	Алгоритмические основы информатики. Алгоритмы, их свойства и средства описания.	Визуализация алгоритмов. Базовые алгоритмические структуры Программы и подпрограммы. Подпрограммы, их назначение и классификация. Оформление подпрограмм, обращение к ним, передача параметров. Библиотечные подпрограммы. Решение задач с помощью стандартных программ	К
3	Основные структуры данных	Перечислимые и диапазонные типы данных. Составные типы. Общая характеристика и классификация. Регулярный тип. Многомерные массивы. Цепочки символов и программные средства их обработки. Комбинированный тип данных. Записи и поля. Записи с вариантами. Оператор присоединения. Множественный тип данных. Понятие конструктора. Операции над множествами. Файловый тип данных и его особенности. Типизированные файлы. Понятие буферной переменной. Работа с файлами последовательного доступа. Логические устройства. Текстовые файлы и их особенности. Типизированные файлы и прямой доступ.	К
4	Средства объектно - ориентированного программирования	Объектно-ориентированная разработка; инкапсуляция и информационное упрятывание; отделение описания поведения от реализации; классы, подклассы и наследование; полиморфизм; иерархия классов; статические и виртуальные методы; динамические объекты; собрания классов и протоколы взаимодействия; программирование на основе шаблонов.	К
5	Линейные динамические структуры	Динамические списки. Однонаправленные и двунаправленные. Стек. Очередь. Дек. Моделирование стека и очереди с помощью массивов, с помощью динамических списков. Деревья. Обходы деревьев. Деревья доичного поиска.	К
6	Структуры данных. Алгоритмы на графах.	Алгоритм определения циклов Алгоритмы планирования процессов Алгоритмы обхода графа Нахождение кратчайшего пути Нахождение минимального остовного дерева	К

К – коллоквиум

### 2.3.2 Занятия семинарского типа.

Занятия семинарского типа не предусмотрены.

### 2.3.3 Лабораторные занятия.

2 семестр

№	Наименование лабораторных работ	Форма текущего контроля
1	2	3
1.	Объекты	Решение задач, отчет по лабораторной работе
2.	Динамические списки	Решение задач, отчет по лабораторной работе
3.	Деревья двоичного поиска	Решение задач, отчет по лабораторной работе
4.	Неориентированные графы	Решение задач, отчет по лабораторной работе
5.	Ориентированные графы	Решение задач, отчет по лабораторной работе

### 2.3.4 Примерная тематика курсовых работ (проектов)

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	Решение индивидуальных задач	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, 2015.-111с., утвержденные кафедрой информационных технологий, протокол № 7 от 09.04.2015 г.
2	Отчет по лабораторной работе	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, 2015.-111с., утвержденные кафедрой информационных технологий, протокол № 7 от 09.04.2015 г.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются 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.

### **3. Образовательные технологии.**

В соответствии с требованиями ФГОС программа дисциплины предусматривает использование в учебном процессе следующих образовательных технологий: 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»	9
2	Л, ЛР	Занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	9
<b>Итого</b>			18

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.**

*Зачетно-экзаменационные материалы для промежуточной аттестации  
(экзамен/зачет)*

##### **Перечень экзаменационных вопросов**

###### **Семестр 1**

1. Алгоритм и его свойства.
2. Машинный язык. Трансляция
3. Функции языков программирования. Основные свойства языков программирования.
4. Основные аспекты изучения языков программирования
5. Метаязыки описания языков программирования
6. Виртуальная машина
7. Типы ошибок, распознаваемые ВМ
8. Структура программы
9. Понятие типа. Структура типов языка
10. Целочисленные типы.
11. Символьный тип
12. Булевский (логический) тип
13. Вещественные типы.
14. Описание переменных и констант
15. Арифметические операции
16. Условная операция. Выражения
17. Поразрядные операции языка
18. Операторы языка. Оператор присваивания
19. Средства ввода-вывода
20. Условный оператор. Вложенность условных операторов
21. Оператор выбора
22. Операторы цикла
23. Операторы перехода
24. Обработка последовательностей. Программирование арифметических циклов
25. Обработка последовательностей. Итерационные циклы
26. Программирование вложенных циклов
27. Конструируемые типы
28. Регулярные типы
29. Обработка одномерных массивов
30. Классы задач по обработке массивов
31. Задачи 1-ого класса. Обоснование выбора структуры данных
32. Задачи 2-ого класса
33. Задачи 3-ого класса
34. Задачи 4-ого класса
35. Методы сортировки. Общая постановка задачи
36. Оценивание алгоритма
37. Метод поиска с обменом (сортировка посредством выбора)
38. Алгоритм «Пузырька»
39. Челночная сортировка (сортировка вставками)
40. Метод подсчета
41. Метод парных сравнений
42. Двумерные массивы
43. Функции
44. Прототип функции
45. Параметры функции. Передача по значению

46. Передача фактических параметров по адресу
47. Локальные и глобальные переменные. Область видимости переменных
48. Параметры со значениями по умолчанию (необязательные параметры)
49. Перегрузка функций
50. Рекурсия
51. Быстрая сортировка
52. Шаблоны функций
53. Указатели
54. Динамические переменные
55. Операции с указателями
56. Ссылки
57. Динамические массивы
58. Передача массивов как параметров в функцию
59. Конструируемые типы
60. Перечисляемые типы
61. Структуры
62. Объединения

#### Семестр 2

1. Информационные структуры. Линейные динамические информационные структуры
  2. Стек. Очередь. Дек.
  3. Моделирование ЛДИС средствами языка. Моделирование стека
  4. Моделирование очереди.
  5. Кольцевой буфер.
  6. Однонаправленные списки. Построение, основные операции.
  7. Двухнаправленные списки. Построение, основные операции.
  8. Деревья
  9. Классы
  10. Ввод-вывод. Система управления вводом-выводом.
  11. Методы доступа. Метки файлов. Открытие и закрытие файлов. Буферизация ввода-вывода
  12. Логические и физические файлы. Общие операции над файлами
  13. Ориентированные графы. Основные определения. Представления ориентированных графов.
  14. АТД для ориентированных графов. Основные операторы.
  15. Задача нахождения кратчайшего пути. Алгоритм Дейкстры.
  16. Нахождение кратчайших путей между парами вершин. Алгоритм Флойда.
  17. Поиск центра ориентированного графа.
  18. Обход ориентированных графов. Поиск в глубину.
  19. Алгоритм нахождения сильно связанных компонент.
  20. Представление неориентированных графов. Основные деревья минимальной стоимости.
- Основные понятия.
21. Алгоритм Прима.
  22. Алгоритм Крускала.
  23. Поиск в ширину на неориентированном графе.
  24. 2-3 деревья. Основные понятия.
  25. Вставка элемента в 2-3 дерево.
  26. Удаление элемента из 2-3 дерева.
  27. Общая схема компиляции.
  28. Информационные таблицы. Сканер.
  29. Семантические подпрограммы. Промежуточные формы представления программ.
  30. Организация таблиц символов.
  31. Задача коллизии. Метод рехеширования.

### 32. Задача коллизии. Метод цепочек переполнения.

*Перечень компетенций, проверяемых оценочным средством:* ОПК-2 – Способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий

#### **Задачи к экзаменационным билетам**

##### **Семестр 1**

###### **Задача 1**

С клавиатуры заполняется массив данных (количество элементов в нем равно  $Lim$ ). Из исходного массива переписать в файл только те элементы массива, сумма четных цифр которых больше заданного числа  $K$ . Вывести результирующий файл. Поиск суммы четных цифр числа оформить в виде функции.

###### **Задача 2**

С клавиатуры заполняется массив данных (количество элементов в нем равно  $Lim$ ). Из исходного массива переписать в файл только те элементы массива, количество пятерок и двоек которых равно заданному числу  $K$ . Вывести результирующий файл. Поиск количества пятерок и двоек в числе оформить в виде функции.

###### **Задача 3**

С клавиатуры заполняется файл данных (количество элементов в нем равно  $Lim$ ). Из исходного файла переписать в массив только те элементы файла, у которых ровно три нечетных делителя. Вывести результирующий массив. Поиск количества нечетных делителей оформить в виде функции.

###### **Задача 4**

С клавиатуры заполняется файл данных (количество элементов в нем равно  $Lim$ ). Из исходного файла переписать в массив только те элементы файла, значение которых не равно максимальному элементу файла и имеет не менее трех четных делителя. Вывести результирующий массив. Поиск количества четных делителей оформить в виде функции.

##### **Семестр 2**

1. Объект 1: поле – одномерный массив; методы: основные методы работы с одномерным массивом; найти наименьший элемент массива, среди элементов, принадлежащий  $[a;b]$ ; найти количество простых элементов.

Объект 2: поле – матрица; методы: ввод-вывод матрицы; найти номера строк, в которых есть хотя бы один простой элемент (наследование); найти сумму наименьших элементов каждой строки, из диапазона  $[-6;8]$ .

Использовать наследование.

2. Построить дерево двоичного поиска. Если в дереве нечетное количество совершенных элементов, то заменить все нечетные элементы их квадратами.

3. Объект 1: поле – одномерный массив; методы: основные методы работы с одномерным массивом; определить является ли элемент совершенным числом; найти сумму элементов, количество цифр в которых не менее 3.

Объект 2: поле – матрица; методы: ввод-вывод матрицы; если в столбце не менее 2 совершенных элементов, то обнулить его; найти номер строки, в которой сумма элементов с количеством цифр не менее 3 максимальна.

Использовать наследование.

4. Построить дерево двоичного поиска. Верно, ли что удвоенное произведение листьев дерева не кратных 5, больше произведения совершенных элементов.

*Перечень компетенций, проверяемых оценочным средством:* ОПК-2 – Способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и

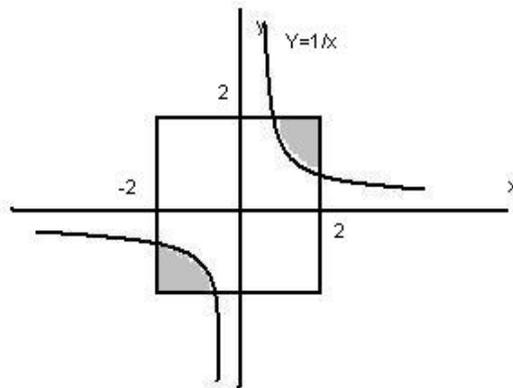
пакеты программ, современные профессиональные стандарты информационных технологий

### Вопросы для коллоквиумов по дисциплине Основы программирования

Семестр 1

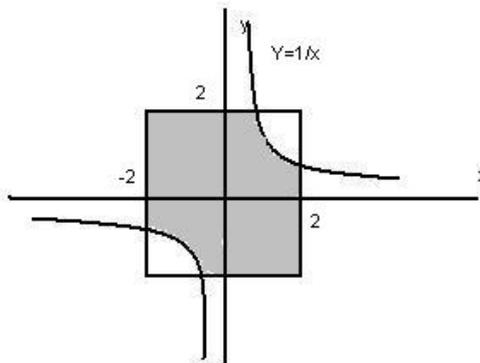
Вариант 1

1. Понятие мета-языка. БНФ. Примеры
2. Оператор выбора. Примеры.
3. Дана квадратная матрица вещественных чисел  $n \times n$ . Найти наибольший по модулю элемент среди элементов, лежащих на побочной диагонали.
4. Дана последовательность целых чисел, оканчивающаяся нулем. Найти количество чисел, сумма цифр которых четна.
5. Дан массив символов. Является ли он симметричным массивом цифр?
6. Пусть  $(x, y)$  – координаты точки на плоскости. Составить булевское выражение, которое принимает значение true тогда и только тогда, когда точка принадлежит заштрихованной области.



Вариант 2

1. Числовые типы.
2. Оператор цикла с постусловием. Синтаксис, семантика. Примеры
3. Дана квадратная матрица целых чисел  $n \times n$ . Заменить нулем элементы с последней цифрой равной 2 среди элементов, лежащих на главной диагонали.
4. Дана последовательность целых чисел, оканчивающаяся -1. Найти среднее арифметическое чисел, количество цифр в которых четно.
5. Дан массив целых чисел. Является ли он упорядоченным по убыванию и содержащим только положительные кратные 3 числа?
6. Пусть  $(x, y)$  – координаты точки на плоскости. Составить булевское выражение, которое принимает значение true тогда и только тогда, когда точка принадлежит заштрихованной области.



## Семестр 2

### Вариант 1

1. Алгоритм сортировки выбором (с обменом)
2. Система управления вводом и выводом файлов
3. Понятие информационной структуры. Динамическая ИС. Очередь.

### Вариант 2

1. Алгоритм сортировки подсчетом
2. Типизированные файлы
3. Понятие информационной структуры. Динамическая ИС. Стек.

*Перечень компетенций, проверяемых оценочным средством:* ОПК-2 – Способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий

## Комплект заданий для контрольной работы по дисциплине Основы программирования

### 1.Тема Обработка массивов

#### Вариант 1

1. Дан массив целых чисел. Найти произведение элементов, в записи которых ровно две цифры 2. Оформить логическую функцию, проверяющую наличие двух цифр 2 в числе. Наличие основной программы обязательно.
2. Дан массив целых чисел. Заменить отрицательные элементы массива на сумму индексов положительных элементов. Оформить рекурсивную функцию вычисления суммы. Оформить рекурсивную процедуру замены элемента на некоторый параметр. Наличие основной программы обязательно.
3. Дан массив строк. Вывести номера несимметричных строк, начинающихся с буквы.
4. Дан массив информации о сотрудниках: фамилия, стаж, зарплата. Найти фамилии сотрудников, заканчивающиеся на «ов», стаж которых число большее заданного числа К, а зарплата трехзначное число.

### 2.Тема Обработка файлов

#### Вариант 1

1. Дан текстовый файл. Найти номера строк, содержащие цифры во второй половине строки.
2. Дан файл записей: фамилия сотрудника, стаж, пол, зарплата. Записать в новый файл фамилии сотрудников, начинающиеся на «А», женщин, с максимальной по файлу зарплатой и стажем – простым числом (использовать функцию).
3. Дан однонаправленный список символов (построить). Если он не упорядочен по возрастанию, то удалить все \*, до которых идет цифра.
4. Дан массив целых чисел. Построить двунаправленный список, содержащий только четные элементы исходного массива. Если список не содержит отрицательных значений, то вставить между двумя элементами, отличающимися не более чем на 2, новый со значением максимального.

### 3.Тема Обработка списков

#### Вариант 1

1. Дано два однонаправленных списка целых чисел. Определить: есть ли вхождение последнего элемента списка 1 в список 2.
2. Дан двунаправленный список. Вычислить  $n * x_1 * x_n + (n-1) * x_2 * x_{n-1} + \dots + 1 * x_n * x_1$ .

*Перечень компетенций, проверяемых оценочным средством:* ОПК-2 – Способность применять в профессиональной деятельности современные языки программирования и

языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий

## **Комплект разноуровневых индивидуальных задач (заданий)** по дисциплине Основы программирования

### **1. Операторы цикла**

1. Дана непустая последовательность различных натуральных чисел, за которой следует ноль. Определить порядковый номер наименьшего из них (while).
2. Дана последовательность из  $n$  целых чисел. Определить, со скольких отрицательных чисел она начинается (for, while).
3. Дано действительное  $x$ . Вычислить приближенное значение бесконечной суммы (сумму первых  $n$  слагаемых):
$$x + \frac{x^2}{2} + \frac{x^3}{3} + \dots \quad (\text{abs}(x) < 1).$$
4. Дано: натуральное число  $n$ , действительные числа  $a_1, \dots, a_n$ . В последовательности  $a_1, \dots, a_n$  определить число соседств двух чисел разного знака.
5. Дано число  $n$ . Определить, является ли оно простым.
6. Дано  $n$  чисел. Определить, сколько из них больше предыдущего и последующих чисел.
7. Дана непустая последовательность натуральных чисел, за которой следует 0. Вычислить сумму тех из них, порядковые номера которых – числа Фибоначчи.

### **3. Одномерные массивы**

1. Даны действительные числа  $a_1, a_2, \dots, a_n$ . Если в результате замены отрицательных элементов последовательности  $a_1, a_2, \dots, a_n$  их квадратами элементы будут образовывать неубывающую последовательность, то получить сумму элементов исходной последовательности, в противном случае каждый пятый элемент заменить средним арифметическим отрицательных чисел.
2. Дан массив целых чисел. Найти среднее арифметическое простых элементов массива.
3. Даны числа  $a_1, a_2, \dots, a_n$ . Переставить элементы последовательности  $a_1, a_2, \dots, a_n$  так, чтобы сначала расположились все неотрицательные элементы, а потом все отрицательные. Порядок как среди неотрицательных элементов, так и среди отрицательных, должен быть сохранен прежним. Затем переставить элементы так, чтобы сначала шли четные элементы, затем нечетные.
4. Даны действительные числа  $a_1, a_2, \dots, a_n$ . Оставить без изменения последовательность  $a_1, a_2, \dots, a_n$ , если она упорядочена по не убыванию или по не возрастанию, в противном случае удалить из последовательности те элементы, порядковые номера которых кратны четырем, сохранив прежним порядок оставленных элементов.
5. Дано: действительные числа  $a_1, \dots, a_n$ ,  $p$ , натуральное число  $k$  ( $a_1 \leq a_2 \leq \dots \leq a_n$ ,  $k < n$ ). Удалить из  $a_1, \dots, a_n$  элемент с номером  $k$  (т.е.  $a_k$ ) и вставить элемент, равный  $p$ , чтобы не нарушилась упорядоченность.
6. Даны действительные числа  $c_1, \dots, c_p, d_1, \dots, d_q$  ( $c_1 \leq c_2 \leq \dots \leq c_p, d_1 \leq d_2 \leq \dots \leq d_q$ ). Внести единую упорядоченность в  $c_1, \dots, c_p, d_1, \dots, d_q$ , получив  $f_1, f_2, \dots, f_{p+q}$  такие, что  $f_1 \leq f_2 \leq \dots \leq f_{p+q}$ . Число сравнений не должно превосходить  $p+q$ .
7. Пусть дан массив  $a_1, \dots, a_n$ . Требуется переставить  $a_1, \dots, a_n$  так, чтобы вначале массива шла группа элементов, больших того элемента, который в исходном массиве

располагался на первом месте, затем – сам этот элемент, потом – группа элементов, меньших или равных ему.

### 3. Регулярные типы. Матрицы

1. Получить целочисленную квадратную матрицу порядка 7, элементами которой являются числа 1, 2, ..., 49, расположенные в ней по спирали.

2. В действительной квадратной матрице порядка  $n$  найти наибольший по модулю элемент. Получить квадратную матрицу порядка  $n-1$  путем выбрасывания из исходной матрицы строки и столбца, на пересечении которых находится найденный элемент.

3. Дана действительная матрица размером  $n \times m$ , в которой не все элементы равны нулю. Получить новую матрицу путем уменьшения всех элементов данной матрицы на ее наибольший элемент.

4. Дано: натуральное число  $m$ , целые числа  $a_1, \dots, a_m$  и целочисленная квадратная матрица порядка  $m$ . Строку с номером  $i$  матрицы назовем отмеченной, если  $a_i > 0$ , и неотмеченной в противном случае. Требуется все элементы, расположенные в отмеченных строках матрицы, преобразовать по правилу: отрицательные элементы заменить на  $-1$ , положительные – на  $1$ , а нулевые оставить без изменения.

5. Дана действительная матрица порядка  $n$ . Заменить нулями все ее элементы, расположенные на главной диагонали и выше нее. Найти среднее арифметическое положительных элементов, расположенных ниже главной диагонали.

6. Дана действительная матрица размера  $m \times n$ . Найти значение наибольшего по модулю элемента матрицы, а также определить есть ли в матрице хотя бы один простой элемент.

7. В данной действительной матрице размера  $m \times n$  поменять местами строку с наибольшим значением, со строкой, содержащей элемент с наименьшим значением. Предполагается, что эти элементы единственны.

### 4. Динамические структуры. 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.

### 5. Двухнаправленные списки. Кольца

1. Дан двухнаправленный кольцевой список целых чисел. Удалить все положительные элементы, до и после которых следуют элементы с нечетной суммой цифр.

2. Дан двухнаправленный кольцевой список целых чисел. Вставить перед каждым элементом, являющимся простым числом, элемент равный количеству цифр этого простого числа.

3. Дан двухнаправленный список целых чисел. Переставить элементы списка так, чтобы сначала следовали элементы со значением меньше среднего арифметического, а затем элементы со значением больше.

4. Дан двухнаправленный 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.

5. Дан двунаправленный список целых чисел. Переставить элементы списка так, чтобы сначала следовали элементы являющиеся числами Фибоначчи, а затем остальные элементы списка.

## 6. Графы

### Задание 1

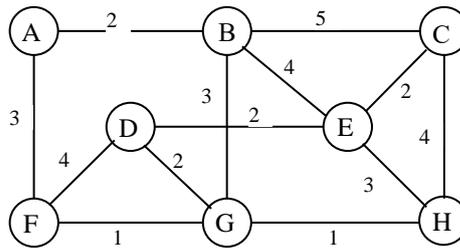
Выполните алгоритм поиска кратчайшего пути из вершины А во все остальные вершины в заданном графе. Подсчитайте количество просмотренных ребер (если обращений несколько, то это нужно учитывать).

$G = \{(a,b,c,d,e,f,g,h),$

$\{a,b\}, \{b,c\}, \{c,h\}, \{h,g\}, \{g,b\}, \{g,f\}, \{f,a\}, \{f,d\}, \{g,d\}, \{d,e\}, \{b,e\}, \{e,c\}, \{e,h\}\},$

стоимости

2, 5, 4, 1, 3, 1, 3, 4, 2, 2, 4, 2, 3



### Задание 2

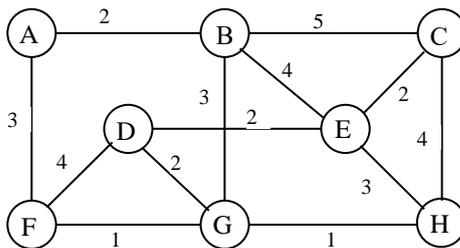
Найдите минимальное остовное дерево с помощью алгоритма Дейкстры-Прима в заданном графе, начиная с вершины А. Выполните анализ алгоритма, подсчитав, сколько раз рассматривается каждое ребро при добавлении вершин к кайме, при обновлении списка ребер ведущих в кайму, и при перенесении вершины из каймы в МОД.

$G = \{(a,b,c,d,e,f,g,h),$

$\{a,b\}, \{b,c\}, \{c,h\}, \{h,g\}, \{g,b\}, \{g,f\}, \{f,a\}, \{f,d\}, \{g,d\}, \{d,e\}, \{b,e\}, \{e,c\}, \{e,h\}\},$

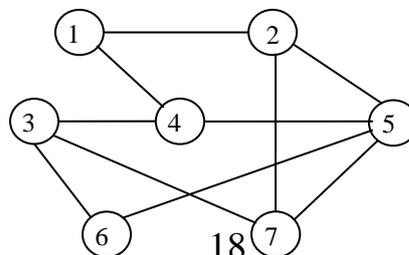
стоимости

2, 5, 4, 1, 3, 1, 3, 4, 2, 2, 4, 2, 3



### Задание 3

Напишите программу, выполняющую обход заданного графа в глубину. Граф представлен с помощью матрицы смежности. Укажите порядок посещения вершин при обходе в глубину начиная с вершины 1.  $G = \{(1,2,3,4,5,6,7), \{1,2\}, \{1,4\}, \{2,5\}, \{5,7\}, \{2,7\}, \{5,4\}, \{5,6\}, \{7,3\}, \{6,3\}, \{4,3\}\}.$



*Перечень компетенций, проверяемых оценочным средством:* ОПК-2 – Способность применять в профессиональной деятельности современные языки программирования и языки баз данных, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий

Компонентом текущего контроля по дисциплине «Основы программирования» являются три контрольные работы в виде письменного решения комплексных задач, предусматривающих знание структур данных.

На контрольной работе каждому студенту дается 4 комплексных задач. Максимальное количество баллов, которое студенты могут получить за правильное решение комплексной задачи на контрольной работе, составляет 2 балла.

Ступени уровней освоения компетенций	Вид задания	Количество баллов
Пороговый	Контрольная работа №1 Обработка массивов Контрольная работа №2 Обработка файлов Контрольная работа №3 Обработка списков	4
Базовый	Контрольная работа №1 Обработка массивов Контрольная работа №2 Обработка файлов Контрольная работа №3 Обработка списков	5-6
Продвинутый	Контрольная работа №1 Обработка массивов Контрольная работа №2 Обработка файлов Контрольная работа №3 Обработка списков	6-8

Каждое решение задачи контрольной работы оценивается в соответствии со следующей таблицей.

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 владеет теоретическими знаниями по структурам данных языка программирования, знает основные алгоритмические конструкции, допускает незначительные ошибки; студент умеет правильно объяснять решение задач.

- **оценка «не зачтено»:** материал не усвоен или усвоен частично, студент затрудняется привести примеры по базовым структурам данных, довольно ограниченный объем знаний по алгоритмическим конструкциям языка программирования, допускает более 4 грубых ошибок при алгоритмическом решении задач.

<b>Оценка</b>	
<b>Незачет</b>	<b>Зачтено</b>
<ul style="list-style-type: none"> <li>• студент получил менее 7 баллов за все контрольные работы</li> <li>• менее 1 балла хотя бы за одну из индивидуальных задач</li> </ul>	<ul style="list-style-type: none"> <li>• студент получил не менее 7 баллов за все контрольные работы</li> <li>• не менее 1 балла за каждую из индивидуальных задач;</li> </ul>

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. 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>

2. Сеницын, С.В. Основы разработки программного обеспечения на примере языка C / С.В. Сеницын, 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>

3. Белоцерковская, И.Е. Алгоритмизация. Введение в язык программирования C++ / 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>

4. 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 [Электронный ресурс]. - 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 [Электронный ресурс]. - 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 [Электронный ресурс]. - 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 [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=364538>

## 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. Основы программирования на языке C++: Учебное пособие/ Сост. С.М. Наместников [http://sernam.ru/lect\\_c.php](http://sernam.ru/lect_c.php)
2. Основы C++. Е. Линский. <https://www.lektorium.tv/course/22825>

## 7. Методические указания для обучающихся по освоению дисциплины.

По курсу предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал; лабораторных занятий, на которых приводятся примеры решений задач по основным учебным темам, выполняются на компьютере с использованием компилятора C++, задачи и упражнения, соответствующие разделам лекционного курса.

Важнейшим этапом курса является самостоятельная работа по дисциплине «Основы программирования».

При самостоятельной работе студентов необходимо изучить литературу, приведенную в перечнях выше, для осмысления 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.

В качестве систем программирования для решения задач и изучения методов и алгоритмов, приведенных в лекциях, рекомендуется использовать на практических занятиях и при самостоятельной работе компилятор C++. Для эффективного программирования рекомендуется использовать встроенные отладчики.

Раздел дисциплины	Форма СР	Сроки выполнения	Формы контроля
Основные понятия языков программирования	Проработка теоретического материала	До 20.10	Коллоквиум
Алгоритмические основы информатики. Алгоритмы, их свойства и средства описания.	Проработка теоретического материала	До 20.10	Коллоквиум
Основные структуры данных	Проработка теоретического материала	До 20.12	Коллоквиум
Средства объектно - ориентированного программирования	Решение индивидуальных задач по темам: Объекты (1)	(1)15.03	Сдача индивидуальных задач
Линейные динамические структуры	Решение индивидуальных задач по темам: Списки (2), Деревья (3)	(2)1.04 (3)15.04	Сдача индивидуальных задач
Структуры данных. Алгоритмы на графах.	Решение индивидуальных задач по темам: Неориентированные графы (4), Ориентированные графы (5)	(4)1.05 (5)15.05	Сдача индивидуальных задач

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.**

- Проверка домашних заданий и консультирование посредством электронной почты.
- Использование электронных презентаций при проведении лекционных занятий.

### **8.2 Перечень необходимого программного обеспечения.**

- Компилятор C++, для разработки программ
- Программное обеспечение для безопасного отображения презентаций

### 8.3 Перечень 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 (ПО), доска
2.	Лабораторные занятия	Лаборатория, укомплектованная техническими средствами обучения – компьютерами с соответствующим программным обеспечением, маркерная доска.
3.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная маркерной доской и оснащенная компьютером.
4.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная техническими средствами обучения – компьютерами с соответствующим 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.