

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кемеровский государственный университет»

Математический факультет



РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.Б.8 Программирование

Направление подготовки
**02.03.02 Фундаментальная информатика и информационные
технологии**

Направленность (профиль) подготовки
Информатика и компьютерные науки

Уровень бакалавриата

Форма обучения
Очная

Кемерово 2015

Рабочая программа дисциплины утверждена Ученым советом факультета
(протокол Ученого совета факультета № 9 от 13. 04. 2015 г.)

Утверждена с обновлениями в части реорганизации структуры факультета
(протокол Ученого совета факультета № 12 от 22. 06. 2015 г.)

Рабочая программа дисциплины одобрена на заседании кафедры ЮНЕСКО по ИВТ

СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы.....	4
2. Место дисциплины в структуре ООП бакалавриата	4
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся	5
3.1. Объем дисциплины (модуля) по видам учебных занятий (в часах).....	5
4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий	6
4.1. Разделы дисциплины (модуля) и трудоемкость по видам учебных занятий (в академических часах).....	6
4.2 Содержание дисциплины (модуля), структурированное по темам (разделам)	8
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю).....	11
6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)	12
6.1. Паспорт фонда оценочных средств по дисциплине (модулю)	12
6.2. Типовые контрольные задания или иные материалы	12
6.3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций	17
7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля).....	18
8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее - сеть «Интернет»), необходимых для освоения дисциплины (модуля)	18
9. Методические указания для обучающихся по освоению дисциплины (модуля).....	19
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости).....	20
11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю).....	20
12. Иные сведения и (или) материалы	21
12.1 Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья	21
12.2 Перечень образовательных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю)	22

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения ООП *бакалавриата* обучающийся должен овладеть следующими результатами обучения по дисциплине (модулю):

<i>Коды компетенции</i>	<i>Содержание компетенций*</i>	<i>результат</i>
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	Знать: современные языки программирования, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий. Уметь: применять в профессиональной деятельности современные языки программирования, методологии системной инженерии, системы автоматизации проектирования, электронные библиотеки и коллекции, сетевые технологии, библиотеки и пакеты программ, современные профессиональные стандарты информационных технологий. Владеть: методикой работы с системной инженерией, электронными библиотеками, сетевыми технологиями, библиотеками и пакетами программ, современными профессиональными стандартами информационных технологий.
ПК–3	— способностью использовать современные инструментальные и вычислительные средства	Знать: современные инструментальные и вычислительные средства. Уметь: использовать современные инструментальные и вычислительные средства. Владеть: способностью использовать современные инструментальные и вычислительные средства.

2. Место дисциплины в структуре ООП бакалавриата

Дисциплина Б1.Б.8 принадлежит базовой учебного плана по направлению подготовки “Фундаментальная информатика и информационные технологии” и является одной из дисциплин, в рамках которой изучаются языки и подходы и к программированию. Курс занимает важное место в профессиональной подготовке специалиста по программированию. Он является одним из основных общепрофессиональных курсов, который лежит в основе изучения других предметов, связанных с программированием и алгоритмизацией. Знания, полученные в результате предмета также необходимы для выполнения курсовых и дипломных работ.

До изучения данного курса студентам необходимы знания в объеме школьного курса информатики и математики

Знания, навыки и умения, приобретенные в результате прохождения курса, будут вос-

требованы при изучении дисциплин специализаций, связанных с программированием таких как численных алгоритмов, так и систем управления базами данных, а также в случае выполнения итоговой квалификационной работы, связанной с реализацией алгоритмов математического моделирования.

Дисциплина изучается на 1-2 курсах в 1,2,3 семестрах

До изучения данного курса студентам необходимы знания в объеме школьного курса информатики и математики. Знания, навыки и умения, приобретенные в результате прохождения курса, будут востребованы при изучении специальных дисциплин, выполнения курсовых и выпускных работ.

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам занятий) и на самостоятельную работу обучающихся

Общая трудоемкость (объем) дисциплины (модуля) составляет **10** зачетных единиц (ЗЕ), **360** академических часов.

3.1. Объем дисциплины (модуля) по видам учебных занятий (в часах)

Объём дисциплины	Всего часов	
	для очной формы обуче- ния	для заочной (очно- заочной) формы обуче- ния
Общая трудоемкость дисциплины	360	
Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	198	
Аудиторная работа (всего):	198	
в том числе:		
Лекции	90	
Семинары, практические занятия		
Практикумы	-	
Лабораторные работы	108	
в т.ч. в активной и интерактивной формах		
Внеаудиторная работа (всего):		
В том числе, индивидуальная работа обучающихся с преподавателем:		
Курсовое проектирование	-	
Групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем		
Творческая работа (эссе)	-	
Самостоятельная работа обучающихся (всего)	162	
Вид промежуточной аттестации обучающегося:		
Зачет (1,2 семестр) - по результатам семинарских занятий	-	
Экзамен(3 семестр)	36	

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкость по видам учебных занятий (в академических часах)

для очной формы обучения

№ п/п	Раздел Дисциплины	Общая трудоемкость в (часах)	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)			Формы текущего контроля успеваемости (по неделям семестра) Форма промежуточной аттестации (по семестрам)
			Учебная работа		Самостоятельная работа	
	всего	лекции	лаб.			

1. Языки и системы программирования

1.1	Информатика как наука. Языки и системы программирования	4	2	2		Устный опрос
1.2	Арифметические основы ЭВМ	7	4	2	1	Устный опрос
1.3	Этапы решения задач на ЭВМ	9	4	4	1	Проверка домашних работ

2. Основные парадигмы программирования

2.1	Структурное программирование	12	4	4	4	Проверка домашних работ
	Вид промежуточного контроля (контрольная работа)	2			2	Контрольной работы
2.2	Модульное программирование	10	2	4	4	Проверка домашних работ
2.3	Использование функций на языке С	12	4	4	4	Проверка домашних заданий

3. Обработка данных

3.1	Составные типы данных: одномерные массивы	12	4	4	4	Проверка домашних работ, устный опрос
3.2	Файлы, текстовые и типизированные файлы	14	4	6	4	Проверка домашних заданий, тест

4. Методы сортировки

	Методы сортировки Сравнение методов сортировки	10	4	4	6	Устный опрос
	Основы доказательства правильности программ. Оценка эффективности про-	8	4	2	2	Устный опрос

	грамм					
	Защита семестровых работ	8			8	Проверка домашних работ
	Итоговый контроль (зачет)					Проверка контрольных работ, устный опрос

5. Составные типы данных

5.1	Составные типы данных: двумерные массивы	20	6	8	6	Устный опрос
5.2	Составные типы данных: строки	20	6	8	6	Проверка семестровых работ
5.3	Структуры	18	6	6	6	Устный опрос

6. Методы решения задач

6.1	Рекурсия как метод решения задач	18	6	6	6	Проверка домашних заданий
6.2	Динамическое программирование	12	6	4	2	Проверка домашних заданий, устный опрос
6.3	Метод ветвей и границ	12	6	4	2	Проверка домашних заданий, устный опрос
	Вид промежуточного контроля (контрольная работа)	2			2	Проверка контрольной работы
	Защита семестровых работ	4			4	Проверка домашних работ
	Итоговый контроль (зачет)					Проверка контрольных работ, устный опрос

7. Объектно-ориентированное программирование

7.1	Развитие языков программирования	2	2			Тестирование по теоретической части
7.2	Основные понятия объектно-ориентированного программирования	16	2	4	2	Тестирование по теоретической части, отчеты по лабораторным заданиям
7.3	Классы, компонентные и дружественные функции, закрытые и открытые компоненты	16	2	4	2	Тестирование по теоретической части, отчеты по лабораторным заданиям
7.4	Наследование, одиночное и множественное. Перегрузка функций	16	2	4	2	Тестирование по теоретической части, отчеты по лабораторным заданиям
7.5	Шаблоны функций и классов, родовые компоненты	1	2	4	2	Тестирование по теоретической части, отчеты по лабораторным заданиям Контрольная работа

8. Абстрактные типы данных

8.1	Концепция типа	4	2	2	4	Тестирование по теорети-
-----	----------------	---	---	---	---	--------------------------

	данных, базовые типы данных					ческой части,
8.2	Абстрактные типы данных, общие понятия	16	2	4	2	Тестирование по теоретической части, отчеты по лабораторным заданиям,
8.3	Абстрактные типы данных: списки, стеки, очереди	28	2	8	4	Тестирование по теоретической части, отчеты по лабораторным заданиям, отчет по семестровой работе.
8.4	Вычисление арифметических выражений	24	2	8	2	Тестирование по теоретической части,
	Подготовка к экзамену	36			36	
	Итоговый контроль (экзамен)					Устный опрос. решение задачи
	Итого	360	90	108	162	

4.2 Содержание дисциплины (модуля), структурированное по темам (разделам)

Содержание лекционных занятий

	Наименование раздела дисциплины	Содержание раздела дисциплины
1. Языки и системы программирования		
.1	Информатика как наука. Языки и системы программирования	История развития языков программирования и вычислительной техники. Основные разделы информатики и задачи, которые решаются. Понятие методологии языков программирования. Обзор основных методологий.
.2	Арифметические основы ЭВМ	Системы счисления. Работа в двоичной и шестнадцатеричной системах счисления. Представление целых чисел в ЭВМ (положительных и отрицательных). Диапазон значений. Представление дробных чисел. Погрешности представления чисел в ЭВМ. Основные типы данных на ЭВМ и диапазоны значений чисел для соответствующих типов данных.
.3	Этапы решения задач на ЭВМ	Поставка задачи. Вопросы, на которые необходимо ответить при постановке задачи. Проектирование. Кодирование. Верификация программы. Понятие спецификации. Пред- и постусловия. Тестирование и отладка программы. Нисходящее и восходящее тестирование программ. Понятие инварианта. Производство и сопровождение программ. Способы записей алгоритмов
2. Основные парадигмы программирования		
.1	Структурное программирование	Кризис программирования, причины его возникновения. Структурный подход в программировании. Основные управляющие структуры. Теорема Бойма и Якопини. Условный оператор на языке С, операторы цикла.
.1	Модульное программирование	Понятие модульного программирования. Принцип утаивания информации, аксиома модульности Коуэна. Основные свойства модуля.
.3	Использование функций на языке С	Структура программы на языке С. Понятие функции. Механизм передачи параметров в функции (по значению, по ссыл-

		ке, с использованием указателей). Локальные и глобальные переменные. Область видимости переменной.
--	--	--

3. Обработка данных

.1	Составные типы данных: одномерные массивы, двумерные массивы, строки	Простые и составные типы данных Понятие массива (одномерный, двумерный). Структуры. Сходства и различия массивов и структур. Работа с одномерными массивами. Основные алгоритмы. Поиск. Максимальный, минимальный элементы. Преобразование массивов.
.2	Файлы, текстовые и типизированные файлы	Типы файлов, функции работы с типизированными и бинарными файлами. Алгоритмы внешней сортировки.

4. Методы сортировки

.1	Методы сортировки Сравнение методов сортировки	Классификация методов сортировки. Простые и улучшенные методы сортировки: сортировка выбором, сортировка вставками, сортировка обменами. Метод Шелла, быстрая сортировка. Параметры по которым производится сравнение методов сортировки. Выбор метода сортировки в различных условиях
.2	Основы доказательства правильности программ. Оценка эффективности программ	Оценка эффективности программ. Понятие эффективности программы. Критерии оценки эффективности программ.

5. Составные типы данных

5.1	Составные типы данных: двумерные массивы	Понятие двумерного массива. Представление двумерных массивов через одномерные. Динамические массивы. Алгоритмы обработки массивов. Преобразование двумерных массивов.
5.2	Составные типы данных: строки	Строки как массив символов. Строки фиксированной и переменной длины. Функции обработки строк.
5.3	Структуры	Составной тип данных – структура. Сходство и отличие с массивом. Массивы структур. Решение задач с использованием структур. Указатели на структуры.

6. Методы решения задач

6.1	Рекурсия как метод решения задач	Рекурсивное определение объектов. Прямая и косвенная рекурсия. Рекурсия как альтернатива итерационным алгоритмам. Классы задач для решением рекурсивным методом. Примеры рекурсивных алгоритмов.
6.2	Динамическое программирование	Динамическое программирование. Классы задач. Решаемые методом динамического программирования. Примеры алгоритмов. Сравнение с другими методами решения. Применимость метода.
6.3	Метод ветвей и границ	Метод ветвей и границ.. Классы задач. Решаемые методом ветвей и границ. Примеры алгоритмов. Сравнение с другими методами решения. Применимость метода

Содержание лабораторных занятий

№	Наименование раздела дисциплины	Содержание раздела дисциплины
1. Языки и системы программирования		
1.1	Информатика как наука. Языки и системы программирования	Введение файловую систему, среда программирования MS Visual Studio
1.2	Арифметические основы ЭВМ	Представление числовой информации в компьютере, Типы данных, их описание, диапазоны значений
1.3	Этапы решения задач на ЭВМ	Разработка программы на языке С, алгоритм, синтаксис программы, отладка. Лабораторная работа № 1. Обработка данных

2. Основные парадигмы программирования		
2.1	Структурное программируемое	Основные алгоритмические структуры Лабораторная работа № 2 Операторы цикла и условия.
2.5	Модульное программирование	Понятие модуля. Стандартные модули на языке С. Подключение модулей.
2.3	Использование функций на языке С	Описание функций. Использование функций для решения задач. Лабораторная работа № 3. Суммирование рядов.
3. Сложные типы данных		
3.1	Составные типы данных: одномерные массивы	Понятие составных типов данных. Работа с одномерными массивами. Лабораторная работа №4. Работа с одномерными массивами
3.2	Файлы, текстовые и типизированные файлы	Работа с файлами различных типов. Ввод-вывод в файлы. Использование файла в различных задачах
4. Методы сортировки		
4.1	Методы сортировки Сравнение методов сортировки	Программирование методов сортировки. Оценка эффективности методов сортировки. Семестровая работа №1. Оценка эффективности методов сортировки
4.2	Основы доказательства правильности программ. Оценка эффективности программ	Оценка сложности алгоритма. Временная сложность. Эффективность использования памяти. Выбор типов данных.
5. Составные типы данных		
5.1	Составные типы данных: двумерные массивы	Понятие двумерного массива Алгоритмы обработки массивов. Преобразование двумерных массивов. Лабораторная работа №5 Семестровая работа: Решение СЛАУ методом Гаусса.
5.2	Составные типы данных: строки	Строки как массив символов. Функции обработки строк. Лабораторная работа №6
5.3	Структуры	Структуры. Решение задач с использованием структур. Лабораторная работа № 7.
6. Методы решения задач		
6.1	Рекурсия как метод решения задач	Решение задач методом рекурсии. Примеры рекурсивных алгоритмов. Лабораторная работа №8 Семестровая работа: нахождение определителя методом разложения по строке.
6.2	Динамическое программирование	Динамическое программирование. Примеры алгоритмов. Решение задач методом динамического программирования.
6.3	Метод ветвей и границ	Метод ветвей и границ Примеры алгоритмов. Решение задач методом ветвей и границ.
7. Объектно-ориентированное программирование		
7.1	Развитие языков программирования	
7.2	Основные понятия объектно-ориентированного программирования	Решение задач с использованием структур.
7.3	Классы, компонентные и дружественные функции,	Лабораторная работа №1 Содержание лабораторной работы:

	закрытые и открытые компоненты	Реализовать класс с набором методов. Написать программу, демонстрирующую работу с объектами. Задания выдаются индивидуально
7.4	Наследование, одиночное и множественное. Перегрузка функций	Лабораторная работа № 2 Содержание лабораторной работы: Реализовать базовый класс и класс-наследник в соответствии с заданием. Определить операции для работы с объектами. Сделать перегрузку операторов вывода, логических и арифметических операторов
7.5	Шаблоны функций и классов, родовые компоненты	Лабораторная работа №3 Содержание лабораторной работы: Создание шаблона класса и реализация на его основе программы, демонстрирующих работу с однотипными объектами, образованными на различных базовых типах.
8. Абстрактные типы данных		
8.1	Концепция типа данных, базовые типы данных	Лабораторная работа №4 Создание атд с набором функций
8.2	Абстрактные типы данных, общие понятия	Лабораторная работа №4 Создание атд с набором функций
8.3	Абстрактные типы данных: списки, стеки, очереди	Лабораторная работа №5 Вычисление алгебраических выражением с использованием стека и очереди. Реализация калькулятора
8.4	Вычисление арифметических выражений	Выполнение семестровой работы по разработке калькулятора

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Дисциплина «Программирование» предполагает как аудиторную (лекции и лабораторные работы), так и самостоятельную работу студентов.

При изучении дисциплины используются следующие материалы учебно-методического обеспечения для самостоятельной работы:

1. Лекции читаются в мультимедийной аудитории. Материал лекций предоставляется обучающимся в форме слайд-конспектов.

2. На практических занятиях проводится опрос, выполняются задания по теме занятия, решаются задачи. Вопросы для подготовки к практическим занятиям представляются студентам в форме текстовых документов

3. В самостоятельную работу студентов входит освоение теоретического материала, , подготовка индивидуальных заданий и проектов, включающих в себя создание программ различной сложности, выполнение проектов по разработке и созданию приложений, подготовка отчетов по лабораторным работам, рефератов.

4. Обучение предполагает выполнение семестровых работ, тематика и требования к которым представлена в методических рекомендациях по выполнению семестровых работ

5. В течение семестра проводится тестовый опрос или проверка индивидуальных заданий.

6. Для подготовки к зачету и экзамену обучающиеся могут воспользоваться конспектом лекций и дополнительными материалами.

7. Зачет проводится в 1-м семестре и предполагает выполнение студентами типовых заданий по пройденным темам.

8. Экзамен проводится по пройденным темам дисциплины во 2 семестре и конце изучения дисциплины (3 семестр)

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине (модулю)

(Перечень компетенций с указанием этапов их формирования; описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания; типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы; методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций)

6.1. Паспорт фонда оценочных средств по дисциплине (модулю)

№	Контролируемые разделы (темы) дисциплины / (результаты по разделам)	Код контролируемой компетенции (или её части) / и ее формулировка – по желанию	наименование оценочного средства
1.	Языки и системы программирования	ОПК-2	контрольное задание, тест
2.	Основные парадигмы программирования	ОПК-2	контрольное задание, защита лабораторной работы , тест
3.	Обработка данных	ОПК-2, ПК-3	контрольное задание, Защита лабораторной работы, тест
4.	Методы сортировки	ОПК-2, ПК-3	Защита семестровой работы, тест
5.	Составные типы данных	ОПК-2, ПК-3	Защита лабораторной работы, защита семестровой работы, тест
6.	Методы решения задач	ОПК-2, ПК-3	Защита лабораторной работы, защита семестровой работы, тест
7.	Объектно-ориентированное программирование	ОПК-2, ПК-3	Защита лабораторной работы. контрольное задание,
8.	Абстрактные типы данных	ОПК-2, ПК-3	Защита лабораторной работы, защита семестровой работы, тест

6.2. Типовые контрольные задания или иные материалы

6.2.1. Экзамен

a) типовые вопросы

1. Определение алгоритма и его основные свойства
2. Языки и системы программирования.

3. Представление чисел в памяти компьютера.
4. Ошибки программирования (ошибки округления).
5. Что включает в себя постановка задачи?
6. Что такое спецификация программы?
7. Примеры спецификаций (например, для метода сортировки массива в возрастающем порядке)
8. Понятие инварианта. Инвариант цикла.
9. Что такое защитное программирование? Перечислите пути реализации защитного программирования.
10. В чем заключаются принципы защитного программирования?
11. Методологии программирования.
12. Модульное программирование: основные понятия.
13. Структурное программирование: основные понятия.
14. Тестирование программ. Восходящее и нисходящее тестирование.
15. Случайные числа. Генераторы случайных чисел.
16. Структура программы на языке С. Арифметические операции. Основные алгоритмические структуры.
17. В сходство и в чем отличия таких понятий как структура и массив?
18. Понятие файла. Сходства и различия с массивами.
19. Классификация файлов. Примеры.
20. Функции работы с файлами.
21. Функции на языке С. Описание, передача параметров, вызов функции.
22. Строковый тип данных. Основные функции работы со строками
23. Рекурсивные объекты и определения. Примеры рекурсивных определений.
24. Рекурсивные функции. Свойства рекурсивного решения.
25. Указатели и основные операции с указателями.
26. Ссылки. Использование ссылок в качестве параметров функции.
27. Концепция типа данных.
28. Типы данных: простые и составные.
29. Эффективность алгоритмов. Подходы к определению.
30. Составные типы данных
31. Двумерные массивы. Способы обработки массивов
32. Динамические массивы
33. Динамическое программирование
34. Метод ветвей и границ
35. Структуры
36. Компонентные функции
37. Объектно-ориентированное программирование – основные понятия (объект, инкапсуляция, наследование, полиморфизм)
38. Классы, компонентные и дружественные функции, закрытые и открытые компоненты
39. Конструкторы и деструкторы

40. Наследование, одиночное и множественное
41. Управление доступом к базовым классам
42. Перегрузка функций
43. Шаблоны функций и классов, родовые компоненты
44. Компонентные функции шаблонного класса
45. Эквивалентность типов, распознавание функции
46. Концепция типов данных
47. Базовые типы данных
48. Сложные типы данных
49. Абстрактные типы данных: (Стеки, Очереди, Списки)
50. Различные реализации абстрактных типов данных (с помощью массивов, списков и т.п.)
51. Вычисление алгебраических выражений

б) типовые задания (контрольное задание)

Пример контрольного задания:

1. При помощи рекуррентных соотношений

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_{n+1}(x) = 2x T_n(x) - T_{n-1}(x), \quad n = 1, 2, \dots$$

составить таблицу значений первых 10 многочленов, принимая $x = 0, 1, 2, \dots, 5$.

Пример контрольного задания:

1. _____ Написать программу вычисления корня уравнения $x = \cos(x + 1)$ методом простой итерации с начальным приближением x_0 и точностью ε . Приближения к корню для уравнения $x = f(x)$ строятся по формуле: $x_k = f(x_{k-1})$, $k = 1, 2, \dots$, до тех пор, пока $|x_k - x_{k-1}| > \varepsilon$.

2. В одномерном массиве, состоящем из n вещественных элементов, вычислить:

- a) максимальный и минимальные элементы массива;
- b) количество элементов массива, лежащих в диапазоне от A до B;

в) типовое задание (тест Test2000)

Примеры тестовых вопросов :

1. Основное синтаксическое понятие методологии императивного программирования это:
 - a. Оператор
 - b. Условный оператор
 - c. Атомарный оператор
 - d. Структурный оператор
2. Согласно аксиомы модульности Коэна, программная единица должна удовлетворять следующим условиям (назовите все верные):
 - a. блочность организации
 - b. параллельное исполнение программ

- c. полнота определения
- d. утаивание информации
- e. синтаксическая обособленность
- f. семантическая независимость
- g. общность данных

3. Результатом выполнения фрагмента программы:

```
int a=4,*u, z;  
u=&z; *u=5;  
a=a+*u+1;  
printf("a= %d", a)
```

будет:

4. Что напечатает следующая программа?

```
y=5;if( z=y<0 ) x=3;  
else if( y==0 ) x=5;  
else x=7;  
print f( “ %d %d ”, x, z )
```

2) Требования к выполнению семестровых работ

Семестровая работа направлена на самостоятельную работу по созданию приложений.

При выполнении семестрового задания необходимо проводить исследование по следующей схеме:

1. Изучить требование к разрабатываемому приложению;
2. Разработать интерфейс приложения;
3. Реализовать все необходимые структуры данных
4. Написать функции для приложения
5. Студент сдает разработанное и реализованное на языке C++ приложение, демонстрирующие основные функции работы с графиками. Приложение должно иметь графический интерфейс.

Семестровые задания даются каждому студенту индивидуально.

3) описание шкалы оценивания

1. Максимальная сумма баллов, набираемая студентом по дисциплине – 100 баллов.
2. В зависимости от суммарного количества набранных баллов, студенту выставляются следующие оценки на экзамене:

0-39 баллов – «неудовлетворительно»;
40-64 баллов – «удовлетворительно»
65-84 баллов – «хорошо»;
85-100 баллов – «отлично».

3. Максимальная сумма баллов промежуточной аттестации (текущей успеваемости) – 70 баллов.

Максимальная сумма баллов семестровой аттестации экзамена) – 30 баллов.

4. Оценка промежуточной аттестации (текущей успеваемости).
 - 4.1. Максимальная сумма баллов промежуточной аттестации (текущей успеваемости) – 80 баллов.

Оценка промежуточной аттестации:

Лабораторные работы (4 работы) – 5 баллов за каждую (всего 20 баллов)

Посещение лекций – 18 баллов

Коллоквиум -15 баллов

Семестровая работа – 15 баллов

Другие виды работ (с ообщение на выбранную тему, выполнение дополнительных заданий и др.) -12 баллов,

- 4.2. Все задания должны быть выполнены на языке С, каждое задание должно сопровождаться набором тестов
- 4.3. Во время защиты лабораторной работы необходимо отвечать на теоретические вопросы полностью.
5. Оценка промежуточной аттестации (зачет).

5.1. Максимальная сумма баллов семестровой аттестации (зачета) – 100 баллов.

Зачет выставляется студенту по результатам работы в семестре. Для получения зачета необходимо выполнить полностью или частично все лабораторные работы, а также сдать и защитить семестровую работу

Некоторые студенты, проявившие активность при изучении курса по усмотрению лектора и преподавателя ведущего лабораторные занятия, которые по итогам текущей аттестации набирают 80 баллов, могут получить бонусные баллы от 1 до 20.

Критерии оценивания теоретических знаний

Раздел 1 Языки и системы программирования

на отметку "зачтено"

- Знать все определения;
- знать основные методологии программирования
- знать этапы разработки программного обеспечения;
- знать устройство памяти компьютера, особенности представления числовой и символьной информации
- знать основные типы данных, обосновывать выбор типов данных;
- Понятие спецификации. Пред- и постусловия.
- Что такое тестирование и отладка программы. Нисходящее и восходящее тестирование программ.
- Понятие инварианта.
- . Способы записей алгоритмов

Раздел 2 Основные парадигмы программирования

на отметку "зачтено" необходимо знать

- 1) Методология императивного программирования
- 2) Структурный подход в программировании
- 3) Основные управляющие структуры. Теорема Бойма и Якопини.
- 4) Условный оператор на языке С, операторы цикла
- 5) Понятие модульного программирования
- 6) Основные свойства модуля.
- 7) Структура программы на языке С.
- 8) Понятие функции. Механизм передачи параметров в функции (по значению, по ссылке, с использованием указателей).
- 9) Локальные и глобальные переменные. Область видимости переменной
- 10) Понятие указатель, основные операции для работы с указателями
- 11) Отличие указателя от ссылки

Раздел 3 Обработка данных

на отметку "зачтено"

- 1) знать понятия простые и составные типы данных ;

- 2) понятие одномерного массива;
- 3) основные методы обработки одномерных массивов;
- 4) методы поиска в одномерных массивах;
- 5) динамические массивы;
- 6) работа с массивами через указатели;
- 7) понятие файла, виды файлов
- 8) отличие файлов от массивов
- 9) функции работы с файлами

Раздел 4 Методы сортировки

- 1) классификация методов сортировки
- 2) простые и улучшенные методы сортировки
- 3) Метод Шелла
- 4) Быстрая сортировка Хоара
- 5) Обоснование методов сортировки
- 6) Оценка эффективности программ
- 7) Критерии эффективности
- 8) Сравнение методов сортировки

;

6.3 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций

Максимальная сумма баллов, набираемая студентом по дисциплине – 100 баллов. Данные баллы студент может набрать регулярно посещая занятия и активно работая на них.

Для получения зачета студенту необходимо полностью или частично сдать все лабораторные работы, коллоквиум и семестровую работу. В случае, если студент не выполнил хотя бы одно из этих заданий. Зачет не выставляется. По прохождении 1 семестра студент должен уметь работать в среде разработки VisualStudio (ПК-3), писать и отлаживать программы (ПК-3), составлять тестовые наборы для проверки правильности работы программы(ПК-3), знать способы обработки одномерных массивов, методы сортировки (ПК-3). В зависимости от суммарного количества набранных баллов в течении семестра, студенту выставляются следующие оценки: 0-59 баллов – «незачтено»; 60-100 баллов – «зачтено».

Студенту, при сдаче теоретического материала на экзамене, необходимо знать основные методологии программирования (ОПК-3), показать свое умение понять поставленную задачу (ОПК-2, ПК-3), знать алгоритмы обработки данных (ОПК-2), осуществлять выбор и анализ соответствующих алгоритмов(ОПК-2, ПК-3), знать основные алгоритмические структуры (ОПК-3), методы решения различных задач (ОПК-2). При сдаче практических заданий, лабораторных и семестровых работ необходимо предоставить программу, написанную на языке С в соответствие с требованиями и ответить на поставленные вопросы. Не сданные во время занятий задания студент может сдать на консультациях.

Соответствие рейтингового показателя традиционной оценке:

Для получения зачета

Рейтинговый балл	Оценка
61-100	Зачтено
Меньше или равно 60	Не засчитано

Для получения экзаменационной оценки

Рейтинговый балл	Оценка
40-64	удовлетворительно
65-84	хорошо
85-100	отлично

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

a) основная литература:

1. Павловская, Татьяна Александровна С/С++. Программирование на языке высокого уровня : для магистров и бакалавров: учебник для вузов / Т. А. Павловская . - СПб. : Питер , 2011
2. Русакова, Н.А. Программирование [Электронный ресурс] : электронный учеб.-метод. комплекс / Н.А. Русакова; Кемеровский гос. ун-т, Кафедра ЮНЕСКО по новым информационным технологиям. - Электрон. дан. - Кемерово : КемГУ, 2009. - 1 эл. опт. диск (CD-ROM) http://unesco.kemsu.ru/study_work/method.htm)
3. Грузина, Э.Э. Практикум по программированию. – Ч. I [Электронный ресурс] : учебное пособие / Э.Э. Грузина, Н.Л. Черноусова. — Электрон. дан. — Кемерово : Издательство КемГУ (Кемеровский государственный университет), 2013. — 100 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=58312
4. Подбельский В.В. Фомин С.С.Курс программирования на языке Си "ДМК Пресс"Издательство: 2012 Год: 384 стр.
http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=4148

б) дополнительная литература:

1. Т. Кормен. и др. Алгоритмы. Построение и анализ алгоритмов- МЦНМО, Москва, 2001.
2. Франка, П. С++ [Текст] : учебный курс: [пер. с англ.] / П. Франка. - СПб. : Питер, 2005. - 521 с
3. Дейл Н. Программирование на С++ [Электронный ресурс] : учебник / Дейл Н., Уимз Ч., Хедингтон М. — Электрон. дан. — М. : ДМК Пресс, 2007. — 672 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=1219

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее - сеть «Интернет»), необходимых для освоения дисциплины (модуля)

Официальные сайты

www.intuit.ru – Национальный Открытый Университете «ИНТУИТ»;

http://e.lanbook.com/books/?p_f_1_temp_id=18&p_f_1_65=917&p_f_1_63=&p_f_1_67= - электронно-библиотечная система, издательство «Лань»;

www.elibrary.ru – научная электронная библиотека;

http://www.edu.ru/modules.php?op=modload&name=Web_Links&file=index&l_op=viewlink&cid=2720 – федеральный портал российского профессионального образования: Математика и естественно-научное образование;

<http://ermak.cs.nstu.ru/trans> - учебные материалы по дисциплине «Теория языков программи-

рования и методы трансляции».

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Данный курс предусматривает изучение теоретических вопросов, в соответствии с рабочей программой, а также выполнение лабораторных и семестровых работ.

К современному специалисту общество предъявляет достаточно широкий перечень требований, среди которых немаловажное значение имеет наличие у выпускников определенных способностей и умения самостоятельно добывать знания из различных источников, систематизировать полученную информацию, давать оценку конкретной финансовой ситуации. Формирование такого умения происходит в течение всего периода обучения через участие студентов в практических занятиях, выполнение контрольных заданий и тестов, написание курсовых и выпускных квалификационных работ. При этом самостоятельная работа студентов играет решающую роль в ходе всего учебного процесса.

9.1. Советы по планированию и организации времени, необходимого для изучения дисциплины. Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины:

Изучение конспекта лекции в тот же день, после лекции – 10-15 минут.

Изучение конспекта лекции за день перед следующей лекцией – 10-15 минут.

Изучение теоретического материала по учебнику и конспекту – 1 час в неделю.

Подготовка к практическому занятию (выполнение домашних заданий) – 1 час.

Всего в неделю – 3 часа 25 минут.

9.2. Описание последовательности действий студента («сценарий изучения дисциплины»).

При изучении дисциплины очень полезно самостоятельно изучать материал, который еще не прочитан на лекции. Тогда лекция будет гораздо понятнее. Однако легче при изучении курса следовать изложению материала на лекции. Для понимания материала и качественного его усвоения рекомендуется такая последовательность действий:

1. После прослушивания лекции и окончания учебных занятий, при подготовке к занятиям следующего дня, нужно сначала просмотреть и обдумать текст лекции, прослушанной сегодня (10-15 минут).

2. При подготовке к лекции следующего дня, нужно просмотреть текст предыдущей лекции, подумать о том, какая может быть тема следующей лекции (10-15 минут).

3. В течение недели выбрать время (1-час) для работы с рекомендованной литературой в библиотеке или ресурсами Интернет.

4. При подготовке к практическим занятиям следующего дня, необходимо сначала повторить пройденный теоретический материал предыдущего занятия по теме домашнего задания. При выполнении упражнения нужно сначала понять, что требуется, какой теоретический материал нужно использовать.

9.3. Рекомендации по использованию материалов учебно-методического комплекса. Рекомендуется использовать методические указания по курсу программирования, текст лекций преподавателя (если он имеется), презентации лекций. Рекомендуется использовать электронные учебно-методические пособия по программированию, имеющиеся на факультетском сервере.

9.4. Рекомендации по работе с литературой. Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта, изучаются и учебники по программированию. Необходимая литература имеется как в библиотеке, так и в кабинете математики. Также по данному курсу имеется достаточно много учебных материалов в электронном виде. При работе с литературой полезно одновременно читать учебники нескольких авторов, после прочтения необходимо выполнить несколько заданий и упражнений самостоятельно, чтобы оценить степень усвоения материала.

9.5. Советы по подготовке к зачету. Дополнительно к изучению конспектов лекции необходимо пользоваться любым рекомендованным учебником по программированию.

При подготовке к зачету нужно изучить необходимый теоретический материал, повторить основные алгоритмы, и самостоятельно решить по нескольку типовых задач из каждой темы.

9.6. Советы по подготовке к экзамену. Дополнительно к изучению конспектов лекции необходимо пользоваться любым рекомендованным учебником по программированию. Необходимо повторить методы решения различных задач, самостоятельно решить часть из них. Внимательно ознакомиться с примерами тестовых заданий.

9.7. Указания по организации работы с контрольно-измерительными материалами, по выполнению домашних заданий. При выполнении домашних заданий необходимо сначала прочитать основные понятия и теоремы по теме задания. При выполнении задания нужно сначала понять, что требуется в задаче, какой теоретический материал нужно использовать, выбрать алгоритм решения задачи, попытаться запрограммировать. Если это не дало результатов, и необходимо рассмотреть решение подобных задач, и после этого попробовать решить предложенную задачу самостоятельно.

Для выполнения домашних работ необходимо использовать информационную систему поддержки учебного процесса (<http://iais.kemsu.ru>). Для доступа к данной системе преподаватель на первом занятии выдает каждому логин-пароль. Домашние задания будут появляться в системе еженедельно после проведения лабораторных занятий. Перед выполнением заданий необходимо повторить пройденный материал, а также изучить рекомендуемую преподавателем литературу для выполнения заданий. Оформление отчета по домашней работе следует выполнить по предлагаемому шаблону (размещен в разделе “Учебные материалы” в системе поддержки учебного процесса).

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

1. Компьютерные классы с набором лицензионного базового программного обеспечения для проведения лабораторных занятий (Microsoft Visual Studio (или CodeBloc) для выполнения лабораторных заданий)
2. Лекционная мультимедийная аудитория для чтения лекций с использованием мультимедийных материалов
3. Тестовая программа Test2000 для компьютерного тестирования;

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Лекционные занятия проходят в мультимедийной аудитории, оснащенной компьютером и проектором. Каждая лекция сопровождается презентацией, содержащей краткий теоретический материал и иллюстративный материал. Каждая презентация построена по следующему шаблону: название лекционного занятия, цель и задачи лекции, краткое содержание предыдущей лекции (при необходимости), теоретический материал (разбит на две части с учетом перемены), в конце приведены итоги лекционного занятия, обозначена тема следующей лекции, а также вопросы и задания для самостоятельного изучения.

Презентации по лекционному курсу разбиты по темам, по отдельно взятой теме может быть несколько лекций.

Лабораторные занятия проходят в компьютерном классе. Первая часть занятия посвящена разбору нового материала. Вторая часть – выполнению практических заданий с целью закрепления материала. При освоении дисциплины для выполнения лабораторных работ необходимы классы персональных компьютеров с приложениями программирования на языках С/С++. Для проведения лекционных занятий, необходима мультимедийная аудитория с набором лицензионного базового программного обеспечения. Основным инструментом для тестирования служит программная среда «Test2000»,

12. Иные сведения и (или) материалы

12.1 Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья

Образование обучающихся с ограниченными возможностями здоровья может быть организовано по выбору студента в следующем виде:

- совместно с другими обучающимися: студент посещает занятия на общих основаниях и непосредственно вовлекается в учебный процесс;
- дистанционно посредством телекоммуникационных технологий: студент прослушивает материал занятий в режиме реального времени, по средствам прямого телемоста (применение Skype или других аналогичных программ и технологий), не находясь непосредственно в учебной аудитории;
- в индивидуальном порядке: преподаватель занимается со студентом индивидуально контактно или посредством телекоммуникационных технологий.

По окончании изучения курса со студентом проводится индивидуальное собеседование, на котором он демонстрирует полученные знания. В случае необходимости, студенту может заранее быть выдано индивидуальное практическое задание, для самостоятельной подготовки (за месяц или за две недели).

Для инвалидов по слуху предусмотрены следующие особенности проведения учебного процесса:

1. Преподаватель предоставляет студенту учебно-методические материалы, необходимые для освоения изучаемого материала (программа курса, план занятия, опорный конспект, методические пособия или слайд презентации, в случае наличия).
2. Лекционный материал преподается в наглядном виде слайд презентаций или сопровождается схемами, наглядными таблицами.
3. Вместо устного ответа студентам предлагается отвечать письменно.
4. Предусматриваются индивидуальные консультации со студентом, на котором может присутствовать сурдопереводчик (университет не обязуется предоставлять сурдопереводчика).

Для инвалидов по зрению предусмотрены следующие особенности проведения учебного процесса:

1. Предусматриваются индивидуальные консультации со студентом, во время которых преподаватель в медленном спокойном темпе объясняет учебный материал (возможно повторно), заостряя внимание на ключевых понятиях.
2. Вместо письменного ответа студентам предлагается отвечать устно.
3. Предлагается ознакомиться с литературой по курсу, написанной шрифтом Брайля, при наличии.

Для инвалидов опорно-двигательного аппарата предусмотрены следующие особенности проведения учебного процесса:

1. Преподаватель предоставляет студенту учебно-методические материалы, необходимые для освоения изучаемого материала (программа курса, план занятия, опорный конспект, методические пособия или слайд презентации, в случае наличия).
2. Лекционный материал преподается в наглядном виде слайд презентаций или сопровождается схемами, наглядными таблицами.
3. Вместо письменного ответа студентам предлагается отвечать устно.
4. Предусматриваются индивидуальные консультации со студентом.

12.2 Перечень образовательных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю)

№ п/ п	Наименование образователь- ной технологии	Краткая характеристика	Представление оценочного сред- ства в фонде
1.	Реферат	Средство, позволяющее оценить умение обучающегося письменно излагать материал по поставленной теме, самостоятельно находить необходимую информацию, анализировать и обобщать ее, делать выводы.	Тематика и требования к реферированию статей и работе со статистическими материалами
2.	Анализ проблемных ситуаций	Метод обучения, предназначенный для совершенствования навыков и получения опыта в следующих областях: выявление, отбор и решение проблем; работа с предположениями и заключениями; оценка альтернатив; принятие решений; слушание и понимание других людей. Позволяет оценить навыки аналитической работы, способность выявлять информацию, необходимую для принятия решений.	Проблемные вопросы.

Составитель: Русакова Н.А. ,доцент кафедры ЮНЕСКО по ИВТ
(фамилия, инициалы и должность преподавателя (ей))

Макет рабочей программы дисциплины (модуля) разработан в соответствии с приказом Минобрнауки России от 19.12.2013 № 1367, одобрен научно-методическим советом (протокол № 8 от 09.04.2014 г.) и утвержден приказом ректора от 23.04.2014 № 224/10..

Макет обновлён с поправками в части подписей на титульной странице, п.3 добавлена строка для указания часов, проводимых в активной и интерактивной формах обучения, добавлен п. 12.1 Особенности реализации дисциплины для инвалидов и лиц с ограниченными возможностями здоровья (протокол НМС № 6 от 15.04.2015 г.), утвержден приказом ректора.