

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М.В.ЛОМОНОСОВА

ФИЛИАЛ В ГОРОДЕ ЕРЕВАНЕ

«УТВЕРЖДАЮ»
Директор Филиала МГУ
имени М.В. Ломоносова в г.Ереване
А.Н.Реймерс
" 09.02.2018 " 2018г.



РАБОЧАЯ ПРОГРАММА

Учебная дисциплина Учебная практика

Направление подготовки Прикладная математика и информатика

Форма обучения очная

Квалификация бакалавр

Курс 1
Семестр 2
Лекции 0
Лабораторные занятия -
Практические занятия -
Семинары -
Форма контроля - зачёт

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Целями дисциплины являются: научить студентов, специализирующихся в области прикладной математики и информатики, пользоваться приобретенными знаниями в области программирования для решения конкретных задач численного и прикладного моделирования.

1.2. Задачи дисциплины:

- ознакомление студентов с базовыми технологиями поддержки процесса разработки программного обеспечения;
- закрепление основ языка программирования Си для решения конкретных задач;
- знакомство с конкретными задачами вычислительной математики.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

А. Информация об образовательном стандарте и учебном плане.

Тип Стандарта: ИБ - интегрированный магистр МГУ, учебный план бакалавриата;

- **направление подготовки** «Прикладная математика и информатика»,
- **наименование учебного плана** ИБ_ПРИКЛАДНАЯ МАТЕМАТИКА И ИНФОРМАТИКА_ФЕ

Б. Место дисциплины в образовательном стандарте и учебном плане:

- практики и научно-исследовательская работа;
- блок дисциплин: Практики
- курс – 1.
- семестр – 2.

В. Перечень дисциплин, которые должны быть освоены для начала освоения данной дисциплины: основы информатики, линейная алгебра и математический анализ I-II.

Г. Общая трудоемкость – 72 часа, 2 з. ед. (2 недели).

3. ПЕРЕЧЕНЬ КОМПЕТЕНЦИЙ, ФОРМИРУЕМЫХ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате изучения данной дисциплины у обучающихся формируются следующие компетенции:

общенаучные:

- владение методологией научных исследований в профессиональной области (ОНК-4);

инструментальные:

- владение навыками использования программных средств и работы в компьютерных сетях, использования ресурсов Интернет; владение основными методами, способами и средствами получения, хранения, переработки информации (ИК-3);

системные:

- способность к творчеству, порождению инновационных идей, выдвижению самостоятельных гипотез (СК-1);
- способность к поиску, критическому анализу, обобщению и систематизации научной информации, к постановке целей исследования и выбору оптимальных путей и методов их достижения (СК-2);
- способность к самостоятельному обучению и разработке новых методов исследования, к изменению научного и научно-производственного профиля деятельности; к инновационной научно-образовательной деятельности (СК-3);

Профессиональные компетенции:

в области научно-исследовательской деятельности:

- способность демонстрации общенаучных базовых знаний естественных наук, прикладной математики и информатики, понимание основных фактов, концепций, принципов и теорий, связанных с прикладной математикой и информатикой (ПК-1);
 - в проектной и производственно-технологической деятельности: способность применять в профессиональной деятельности современные языки программирования и методы параллельной обработки данных, операционные системы, электронные библиотеки и пакеты программ, сетевые технологии (ПК-3);
 - способность собирать, обрабатывать и интерпретировать экспериментальные данные, необходимые для проектной и производственно-технологической деятельности (ПК-5);
- в организационно-управленческой деятельности:*
- способность приобретать и использовать организационно-управленческие навыки в профессиональной и социальной деятельности (ПК-7);
- в инновационной деятельности:*
- способность приобретать новые научные и профессиональные знания, используя современные образовательные и информационные технологии (ПК-10);
 - способность осуществлять целенаправленный поиск информации о технологических достижениях в сети Интернет и из других источников (ПК-11);
- в научно-педагогической деятельности:*
- способность владения методикой преподавания учебных дисциплин (ПК-12);
 - умение работать самостоятельно и в коллективе, руководить людьми, разъяснять и самостоятельно выполнять порученные задания (ПК-14).

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины составляет 2 зачетных единиц или 72 ч., из них:

- лекции – 0 часов;
- семинары – 0 часов;
- лабораторная работа – 0 часов;
- самостоятельная работа – 72 часа.

Форма контроля – зачёт.

4.1. Распределение трудоемкости по разделам и темам, а также формам проведения занятий с указанием форм текущего контроля и промежуточной аттестации:

№ п/п	Наименование разделов (этапов) практики	Трудоемкость (в ак. часах) по формам занятий (для дисциплин) и видам работ (для практик)	Формы контроля
		<i>Самостоятельная работа</i>	
1	Раздел 1. Среда UNIX и базовые конструкции языка Си		
	Тема 1. Рабочая среда UNIX, базовые команды и утилиты командной оболочки UNIX .	6	Собеседование.
	Тема 2. Работа с SSH и системой контроля версий git	2	Собеседование.
	Тема 3. Язык Си (типы, операции, декларация переменных, функции: передача параметров, операторы перехода, блоки	8	Собеседование.

	операторов и области видимости переменных)		
	Тема 4. Препроцессор языка Си	2	Собеседование.
	Тема 5. Стандартная библиотека языка Си	4	Собеседование
	Тема 6. Язык Си, как язык раздельной трансляции, опции компиляции, этапы работы компилятора	4	Собеседование
	Тема 7. Работа с динамической памятью, отладка обращений в память с использованием valgrind	6	Собеседование
	Тема 8. Работа со средствами разработки в UNIX (отладка с использованием gdb, язык Makefile-ов, текстовый редактор vim)	8	Собеседование
2.	Раздел 2. Выполнение практических заданий		
	Тема 9. Выполнение заданий на организацию ввода/вывода в программе и передачу параметров	4	Составление и защита отчета
	Тема 10. Выполнение заданий на работу со строками, и динамической памятью, алгоритмы сортировки	6	Составление и защита отчета
	Тема 11. Реализация математических алгоритмов, требующих работы с динамической памятью и строками	10	Составление и защита отчета
	Тема 12. Реализация математических алгоритмов, требующих сложных структур данных.	12	Составление и защита отчета
3	Итого	72	зачёт

4.2. Содержание практики по разделам и темам (этапам) - самостоятельная работа:

Примеры заданий практикума

1. Процент частоты встречи символов в тексте

На стандартный поток ввода программе поступает текст. Программа должна считывать текст до тех пор, пока поток ввода не будет закрыт (например одновременным нажатием клавиш <ctrl> и <d> в терминале). Требуется распечатать процент встречаемости символа в тексте. Если символ не отображаем на экране, то вместо изображения символа в терминале необходимо напечатать его код в десятичной системе исчисления.

2. Поиск нулей многочлена на отрезке методом деления отрезка пополам

В текстовом файле хранятся коэффициенты (как числа с плавающей точкой в формате double) при степенях многочлена в порядке возрастания степеней. Требуется написать программу вычисляющую корни данного многочлена на заданном отрезке с заданной точностью. Программе на вход передаются границы отрезка, точность, предельное число итераций. Программа должна распечатывать найденные корни уравнения, а также проиллюстрировать процесс приближения к корню.

3. Усложнённый метод деления отрезка пополам

Условия задания аналогичны предыдущему. Но предполагается, что что мы область поиска нулей функции дополнительно разделим на n -частей, в каждой из которых будем независимо искать корень многочлена методом деления отрезка пополам. В дальнейшем необходимо будет производить процедуру отождествления идентичности корней, если истинный корень оказался вблизи границ отрезков.

4. Расчет площади методом прямоугольников, при неравномерном разбиении области интегрирования

Предполагается, что в некотором .c файле описана функция с прототипом:
double f(double x);

На вход программе передаются границы отрезка интегрирования, число точек — n , на которое нужно дробить отрезок, а так же ε . В дальнейшем каждый частичный отрезок может быть разбит ещё на n частей, в случае если выполняется условие: $|f(x_i) - f(x_{i+1})| \geq \varepsilon$. Множество получившихся раздробленных интервалов можно хранить в динамическом массиве, как массив структур (left,right,delta_x). После проведения процедуры дробления интервалов требуется вычислить интеграл распечатать, все частичные площади и интервалы.

5. Перемножение плотных матриц

Требуется написать 2 программы. Программу генератор: создаёт файл с матрицей действительных чисел. Матрица хранится в формате: сперва идёт слово Matrix, далее запись в текстовом виде как два десятичных целых разделённых пробельными символами размерность матрицы число строк и число столбцов. Далее через пробельные символы следует запись чисел с плавающей точкой, типа double. Вторая программа перемножает две матрицы, имена файлов с матрицами передаются программе в аргументах функции main, и распечатывающую результат в стандартный поток вывода (Если в процессе выполнения программы возникли ошибки, то они должны быть выведены в стандартный поток ошибок).

6. Перемножение разреженных матриц

Требуется написать 2 программы: генератор и программу для перемножения матриц. Однако в данном задании матрица хранится в другом формате, не в таком как в предыдущем задании. В начале идёт слово Sparse_matrix, затем числа задающие размерность матрицы. Далее список координат в матрице и значение в них. Предполагается, что в файле хранятся только не нулевые значения матрицы. В оперативной памяти также должны храниться только не нулевые элементы. Предполагается, что в оперативной памяти матрица представлена массивом структур (i, j, значение) Вывод на печать результата перемножения матриц, так же должен быть в разреженном формате, где каждая тройка (i, j, значение) должны быть написаны на новой строчке.

7. Усложнённое перемножение матриц

Условия аналогичные предыдущему заданию, однако предполагается, что для облегчения поиска значений в матрице, в оперативной памяти строится дерево индексов, как двоичное дерево поиска по координатам в матрице. С этой целью можно использовать либо полностью сбалансированные деревья, либо красно-чёрные деревья, либо декартовы деревья.

8. Вычисление обратной матрицы методом Гаусса

Для разреженных матриц, представленных в файле как в задании “перемножение разреженных матриц” требуется создать программу, которая записывает в файл обратную матрицу, а также распечатывает на экран значение определителя исходной матрицы. Для этого матрицу нужно привести к треугольному виду применяя метод гаусса, но сохраняя значения на диагонали. В памяти элементы матрицы должны храниться как список (для предоставления возможности удаления нулевых элементов матрицы, которые обнулятся в процессе вычисления). В программе необходимо предусмотреть проверку правильности обратной матрицы. В результате вычисления, должна получиться матрица, где элементы не более, чем на epsilon отличаются от 1 - для диагональных элементов и, от 0 - для остальных элементов матрицы.

9. Кластеризация текстов

На вход программе подаётся текстовый файл, где на каждой строчке указано имя файла. Для каждого из файлов необходимо подсчитать частоты встречаемости символов. Далее для каждой пары текстов вычисляется метрика расстояния между текстами на основе частот. Например можно вычислить вот такую метрику: $\frac{1}{256} \sum_{i=0}^{255} |x_i - y_i|$, здесь x_i - частота символа с кодом i в первом тексте, а y_i - частота во втором тексте.

Далее, все такие расстояния собираются в матрицу. Матрица должна иметь формат разреженной матрицы, где элементы матрицы хранятся в списке. Расстояния большие некоторого порога выкидываются из матрицы. Далее, данная матрица интерпретируется как неориентированный граф, где индексы i, j задают ребро, а значение матрицы задаёт “вес ребра”. В результате удаления рёбер из исходного полного графа, граф теперь может оказаться не связанным. Требуется найти все связанные компоненты в данном графе, и затем напечатать список файлов оказавшихся в компоненте (список файлов соответствует набору вершин в графе, где каждый файл однозначно сопоставлен вершине графа). Каждая компонента связности должна быть сохранена в своём файле.

10. Реализация длинной целочисленной арифметики

Требуется создать библиотеку функций с операциями сложить, вычесть, умножить, поделить нацело, взять остаток от деления. В том числе: чтение из файла, запись в файл, распечатка числа в стандартный поток вывода. Для тестирования библиотеки написать программу, где числа задаются в 2-х файлах. Число удобно представлять как структуру хранящую знак, длину числа, само число, в виде строки задом наперёд (младшие разряды на первых позициях строки).

11. Рациональные длинные числа

В файле рациональные числа записаны в виде дроби вида: «-123123123/777» либо «234523452345/14». При этом если число целое, то оно представляется в файле стандартным образом (символ '/' и часть после него отсутствуют). Требуется, так же, как и для целых чисел реализовать операции сложить, вычесть, умножить, поделить, а также операцию приведения рационального числа к несократимой дроби. Для этого реализовать любой алгоритм факторизации целых чисел (можно не эффективный).

12. Метод гаусса для решения системы линейных алгебраических уравнений в длинных рациональных числах

На вход программе подаётся матрица рациональных чисел, а так же вектор f с правой частью уравнения $Ax = f$. Матрица записывается также как в задании «перемножение плотных матриц», с той разницей, что вместо чисел с плавающей точкой представлены рациональные числа в длинной арифметике. Требуется сообщить ранг матрицы, а так же напечатать в стандартный поток вывода решение уравнения, если оно единственно.

5. ИСПОЛЬЗУЕМЫЕ ОБРАЗОВАТЕЛЬНЫЕ, НАУЧНО-ИССЛЕДОВАТЕЛЬСКИЕ И НАУЧНО-ПРОИЗВОДСТВЕННЫЕ ТЕХНОЛОГИИ

А. Образовательные технологии: активные и интерактивные формы, практические занятия, самостоятельные работы. В течение практики студенты решают задачи, указанные преподавателем, к каждому семинару. В каждом семестре проводятся проверки выполнения домашней работы.

Б. Научно-исследовательские технологии: решение задач практического характера, отладка и написание программ.

В. Научно-производственные технологии: решение задач прикладного и практического характера, выявление возможности и метода применения полученных знаний в других разделах математики и ее приложениях.

6. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ.

А. Учебно-методические рекомендации для обеспечения самостоятельной работы студентов: разбор домашних заданий.

Б. Оценочные средства контроля успеваемости:

Оценка практики выставляется студенту в зависимости от количества решённых заданий и сложности решённых заданий. Например, цепочка заданий 6-10-11-12 означает оценку отлично (4 – технических балла); 6-10-11 — хорошо (3 - балла) или отлично (4-балла) в зависимости от выбранного метода факторизации чисел; 6-10 — удовлетворительно (2 - балла) или хорошо (3 - балла), в зависимости от эффективности реализации операций умножения и деления, цепочка заданий, состоящая только из задания 6 — означает, что студент получает (1 – технический балл). Для получения оценки “зачтено” за всю практику необходимо получить технический балл не менее 2-х. В случае меньшего технического балла выставляется оценка “не зачтено”, так как не набрана минимальная сложность заданий, не охвачены темы практики «Реализация математических алгоритмов, требующих работы с динамической памятью и строками» и «Реализация математических алгоритмов, требующих сложных структур данных».

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

7.1. Основная литература:

	Автор	Название книги/статьи	Место издания	Издательство	Год издания
1	Томас Х. Кормен, Чарльз И. Лейзерсон, Рональд Л. Ривест, Клиффорд Штайн	Алгоритмы: построение и анализ, 3-е издание Introduction to Algorithms, Third Edition 1328 с.	Москва	Вильямс	2013
2	Калиткин Н.Н.	Численные методы 592 с.	Спб.	БХВ-Петербург	2011
3	Брайан Керниган, Деннис Ритчи	Язык программирования С. 304 с.	Москва	Вильямс	2015
4	Столяров А.В.	<i>Операционная среда ОС UNIX для изучающих программирование.</i> (Архитектура ЭВМ и СПО. Пособие для выполнения лабораторных работ в ОС UNIX)	Москва	МГТУГА	2008
5	Курячий Г. В., Маслинский К. А.	Операционная система Linux. — ALT Linux. С. 347.	Москва		2010

7.2. Дополнительная литература:

	Автор	Название книги/статьи	Место издания	Издательство	Год издания
1	Дейв Тейлор, Брендон Перри	Сценарии командной	Петербург	СПб.	2017

		оболочки. Linux, OS X и Unix. 416 с.			
2	Фленов М. Е.	Linux глазами хакера, 4-е изд. 432 стр.	БХВ-Петербург		2017
3	Чакон С., Штрауб Б.	Git для профессионального программиста. 496 с.	Петербург	СПб	2016
4	Альфред В. Ахо, Джон Э Хопкрофт, Джеффри Д. Ульман	Структуры данных и алгоритмы. 400с.		Вильямс	2016
5	Гасфилд	Строки, деревья и последовательности в алгоритмах	СПб.	Невский Диалект, БХВ-Петербург	2003
6	Кнут Д.	Искусство программирования. Том 3. Сортировка и поиск. Второе издание. 824 с.		Вильямс	2017

7.3. Программное обеспечение:

Microsoft Windows Pro 7 (США), Microsoft Office Pro + 2010 64 (США).

Установленная UNIX-подобная операционная система; на каждом компьютере должно быть установлено следующее программное обеспечение:

1. текстовый редактор vim
2. клиент ssh
3. система контроля версий git
4. компилятор языка Си gcc
5. отладчик программ gdb
6. отладчик обращений программы к виртуальной памяти valgrind
7. Xserver и одна или несколько графических оболочек (например xfce или KDE)
8. программа astyle — для приведения исходных кодов к определённому формату.
9. Установленный на уделённой машине сервер репозитория.

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

А. Помещения: учебная аудитория.

Б. Оборудование: компьютерный класс (с необходимым материальным обеспечением)

В. Иные материалы: для демонстраций в аудитории требуется проектор и доска.

Программа соответствует ОС МГУ имени М.В. Ломоносова и учебному плану направления подготовки 01.03.02. «Прикладная математика и информатика».

Разработчик:

Сальников А.Н. к.ф.-м.н, ведущий научный сотрудник факультета ВМК МГУ



Одобрена на заседании Ученого Совета Филиала МГУ имени М.В. Ломоносова в г.

Ереване 20.01.2018 г, протокол № 1