

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО «СГУ имени Н.Г. Чернышевского»

Факультет компьютерных наук и информационных технологий

УТВЕРЖДАЮ

«13» 01



**Рабочая программа дисциплины
Методы программирования**

Специальность
10.05.01 Компьютерная безопасность

Специализация
Математические методы защиты информации

Квалификация выпускника
Специалист по защите информации

Форма обучения
очная

Саратов,
2017

1. Цели освоения дисциплины «Методы программирования»

Целями освоения дисциплины являются: знакомство с основными комбинаторными и теоретико-графовыми алгоритмами, а также способами их эффективной реализации и оценки сложности, знакомство с современными технологиями программирования.

2. Место дисциплины «Методы программирования» в структуре ООП

Данная учебная дисциплина относится к базовой части Блока 1 «Дисциплины (Модули)» ООП и направлена на формирование у обучающихся общепрофессиональных компетенций.

Для изучения дисциплины необходимы компетенции, знания, умения и готовности, сформированные у обучающихся в результате освоения курсов «Дискретная математика», «Математическая логика и теория алгоритмов», «Языки программирования».

Компетенции, знания, умения и навыки, сформированные у обучающихся в результате освоения данной дисциплины, могут быть использованы для выполнения курсовых работ.

3 Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен обладать следующими общепрофессиональными компетенциями:

- способностью применять методы научных исследований в профессиональной деятельности, в том числе в работе над междисциплинарными и инновационными проектами (ОПК-4);

- способностью использовать языки и системы программирования, инструментальные средства для решения профессиональных, исследовательских и прикладных задач (ОПК-8);

- способностью к самостоятельному построению алгоритма, проведению его анализа и реализации в современных программных комплексах (ОПК-10).

В результате освоения дисциплины обучающийся должен

Знать:

- основные комбинаторные и теоретико-графовые алгоритмы, а также способы их эффективной реализации и оценки сложности;
- современные технологии программирования;
- показатели качества программного обеспечения;
- архитектуру основных типов современных компьютерных систем;
- основы системного программирования;
- принципы и методы построения быстрых алгоритмов для реализации систем защиты информации;

- основные алгоритмы кодирования, сжатия и восстановления различных видов информации;

Уметь:

- проводить оценку сложности алгоритмов;
- разрабатывать эффективные алгоритмы и программы;
- планировать разработку сложного программного обеспечения;
- оценивать качество готового программного обеспечения;
- формализовать поставленную задачу;
- работать с интегрированными средами разработки программного обеспечения;
- разрабатывать системное и прикладное программное обеспечение для многозадачных, многопользовательских и многопроцессорных сред, а также для сред с интерфейсом, управляемым сообщениями;
- разрабатывать и сопровождать программные средства с учетом требований к их защищенности;
- проводить предварительное оценивание временной сложности разрабатываемых алгоритмов;
- разрабатывать быстрые алгоритмы для криптографических приложений;

Владеть:

- навыками разработки алгоритмов решения типовых профессиональных задач;
- навыками разработки, документирования, тестирования и отладки программ;
- навыками использования инструментальных средств разработки и отладки программного кода;
- навыками анализа вычислительной сложности разработанных алгоритмов;
- методами разработки эффективных алгоритмов;
- базовыми алгоритмами и структурами данных.

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 9 зачетных единиц 324 часа.

№ п/п	Раздел дисциплины	С е м е с т р	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)				Формы текущего контроля успеваемости (по неделям семестра) Формы промежуточной аттестации (по семестрам)
				Лек	Пр	Лаб	Сам	
1	2	3	4	5	6	7	8	9
5 семестр								
1	Введение	5	1	2			12	
2	Алгоритмы сортировки	5	2-5	8		6	12	
3	Алгоритмы эффективного поиска	5	6-11	12		18	12	Контрольная работа №1
4	Алгоритмы вычислительной геометрии	5	12-13	4		4	12	
5	Хеш-функции, хеш-таблицы	5	14-15	4		4	12	
6	Алгоритмы теории строк	5	16-18	6		4	12	
	Промежуточная аттестация	5						Зачет
	Итого	5		36		36	72	
6 семестр								
1	Графы, графовые алгоритмы	6	1-7	14		18	11	Контрольная работа №2
2	Алгоритмы на перестановках	6	8	2		2	12	
3	Динамическое программирование	6	9-11	6		8	12	
4	Параллельные алгоритмы	6	12-13	4		2	12	
5	Современные технологии программирования	6	14	2		0	12	
6	Объектно-ориентированное программирование	6	15-16	4		2	12	
	Промежуточная аттестация	6						Экзамен
	Итого	6		32		32	71	45
	ВСЕГО			68		68	143	45

Раздел «Введение»

Понятие «алгоритм», примеры. Временная сложность алгоритма. Значение тета-, омега-, о- нотаций (большие и малые). Свойства тета-, омега-, о- нотаций.

Раздел «Алгоритмы сортировки»

Задача о сортировке. Виды сортировок. Теорема о нижней оценке числа сравнений для сортировок. Сортировка слиянием. Структура данных «куча». Основные операции с кучей. Очереди с приоритетами. HeapSort. Быстрая сортировка. Общий вид быстрой сортировки. Варианты выбора опорного элемента. Операция partition. Анализ времени работы и используемой памяти. Сравнение с HeapSort. Алгоритм нахождения k-ой порядковой статистики. Линейные сортировки. CountSort. RadixSort. Примеры работы алгоритмов. Приложения сортировок: задачи на сортировки.

Раздел «Алгоритмы эффективного поиска»

Бинарный поиск. Назначения и идея алгоритма. Оценка количества действий. Функции стандартной библиотеки C++, основанные на бинарном поиске. Модификации: реализация `index_first_of`, `index_last_of` на основе стандартной библиотеки C++. Задача о `minimax` пути в графе. Реализации функций `binary_search`, `index_first_of`, `index_last_of` без использования функций стандартной библиотеки C++, основанных на бинарном поиске. Оптимизация бинарного поиска для лучшего использования кэша процессора. Вещественнозначный бинарный поиск. Троишный (тернарный) поиск. Назначение и применимость. Реализация. Время работы. Приложения.

Раздел «Алгоритмы вычислительной геометрии»

Скалярное и векторное произведения векторов. Физический смысл и формулы вычисления. Примеры приложений: проверка взаимного расположения четырех точек, проверка многоугольника на выпуклость. Проверка пересечения двух отрезков. Алгоритмы нахождения выпуклой оболочки. Обзор методов. Просмотр Джавриса. Обход Грэхема. Задача о двух наиболее удаленных точках. Задача о нахождении пары ближайших точек.

Раздел «Хеш-функции, хеш-таблицы»

Хеш-функции. Виды хэш-функций (по области применения). Особенности каждого из видов. Хэш-таблицы. Основные операции с хэш-таблицами. Время работы хэш-таблиц. Методы разрешения коллизий.

Раздел «Алгоритмы теории строк»

Задача поиска образца в тексте. Два вида препроцессинга данных. Префикс-функция строки. Алгоритм нахождения префикс функции строки. Анализ, доказательство корректности. Алгоритм Кнута-Морриса-Пратта. Доказательство корректности и анализ. Суффиксные деревья. Простейшие алгоритмы построения суффиксных деревьев. Примеры.

Раздел «Графы, графовые алгоритмы»

Графы. Основные определения. Формулировка задачи SSSP. Дерево кратчайших путей. Свойства. Алгоритм нахождения кратчайшего пути из s в

t по дереву кратчайших путей из вершины s. Алгоритм BFS. Его анализ и свойства. Постановка задачи SSSP на взвешенных графах. Случаи возможных постановок задачи. Алгоритм Дейкстры. Инварианты выполнения алгоритма Дейкстры. Доказательство корректности. Анализ времени работы. Алгоритм Форда-Беллмана. Обоснование корректности, время работы. Алгоритм Флойда. Обоснование корректности, время работы. Условия применимости каждого из алгоритмов. Сравнение различных алгоритмов решения задачи SSSP. Краткое описание каждого из них, применимость, время работы, особенности. Примеры задач, сводящихся к SSSP. Алгоритм поиска в глубину. Лемма о скобочном вложении. Лемма о сером пути. Примеры работы алгоритма. Мосты и точки сочленения. Топологическая сортировка.

Раздел «Алгоритмы на перестановках»

Перестановки. Перестановки как преобразования. Произведение перестановок. Обратная перестановка. Группа перестановок. Алгоритмы перемножения и нахождения обратной перестановки. Граф перестановки. Порядок перестановки. Нахождение номера перестановки в лексикографическом упорядочении. Нахождение перестановки по ее номеру в лексикографическом упорядочении. Подсчет числа инволюций. Таблица инверсий перестановки. Алгоритм восстановления перестановки по таблице инверсий и наоборот.

Раздел «Динамическое программирование»

Динамическое программирование. Идеология. Оптимальность подзадач. Примеры.

Раздел «Параллельные алгоритмы»

Многопроцессорные системы. Средства описания параллельных алгоритмов; методы построения параллельных алгоритмов; параллельные алгоритмы сортировки данных; параллельные алгоритмы генерации псевдослучайных чисел.

Раздел «Современные технологии программирования»

Тенденции в мире программирования. Парадигмы программирования. Концепции языков программирования. Разработка через тестирование. Switch-технология.

Раздел «Объектно-ориентированное программирование»

История. Основные понятия. Определение ООП и его основные концепции. Диаграммы UML. Особенности реализации. Критика ООП. Объектно-ориентированные языки.

План лабораторных занятий

		Лаб	Сам	Задания
	5 семестр			
1.	Введение		12	Просмотр функциональных возможностей сред разработки ПО
2.	Алгоритмы сортировки	6	12	Реализация сортировок пузырьком, выбором, вставками, кучей, быстрая сортировка, линейные сортировки; решение задач, использующих сортировку как часть алгоритма
3.	Алгоритмы эффективного поиска. Бинарный поиск	18	12	Реализация методов <code>index_first_of</code> , <code>index_last_of</code> на основе стандартной библиотеки C++ и без использования функций стандартной библиотеки C++. Решение задач при помощи бинарного и тернарного поиска
4.	Алгоритмы вычислительной геометрии	4	12	Реализация алгоритмов нахождения выпуклой оболочки. Задача о двух наиболее удаленных точках. Задача о нахождении пары ближайших точек.
5.	Хеш-функции, хеш-таблицы	4	12	Реализация хэш-функции, реализация методов обработки коллизий
6.	Алгоритмы теории строк	4	12	Реализация алгоритмов: Кнута-Мориса-Пратта, построения суффиксных деревьев
	6 семестр			
1.	Графы, графовые алгоритмы	18	11	Реализация алгоритмов: BFS, Дейкстры, Флойда, Форда-Беллмана, поиска в глубину, топологической сортировки
2.	Алгоритмы на перестановках	2	12	Реализация алгоритма восстановления перестановки по таблице инверсий и наоборот
3.	Динамическое программирование	8	12	http://informatics.mccme.ru/moodle/mod/statements/view.php?id=269 Система автоматической проверки задач
4.	Параллельные алгоритмы	2	12	Реализация параллельного алгоритма сортировки данных, генерации псевдослучайных чисел.
5.	Современные технологии программирования	0	12	Создание UNIT-тестов для уже созданных программ
6.	Объектно-ориентированное программирование	2	12	Реализация предыдущих задач, используя принципы ООП

5. Образовательные технологии

В рамках лекционных занятий предполагается разбор различных задач и ситуаций, обсуждение в малых группах, обсуждение с привлечением всей аудитории.

Для поддержки обучения студентов по данному курсу используется технология смешанного обучения. Закрепление практических навыков решения задач по программированию происходит с последующей сдачей решений в автоматизированную систему проверки.

При обучении лиц с ограниченными возможностями здоровья и инвалидов используются подходы, способствующие созданию безбарьерной образовательной среды: технологии дифференциации и индивидуализации обучения, сопровождение тьюторами в образовательном пространстве; увеличивается время на самостоятельное освоение материала.

Основной формой организации учебного процесса является интегрированное обучение лиц с ограниченными возможностями здоровья и инвалидов, т.е. все студенты обучаются в смешенных группах, имеют возможность постоянно общаться со сверстниками, благодаря чему легче адаптируются в социуме.

6. Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины.

Виды самостоятельной работы студентов

- 1) Изучение дополнительной литературы по предмету. При чтении лекций по соответствующим разделам дисциплины даются ссылки на источники, в которых более детально рассматривается материал.
- 2) Решение задач по программированию с последующей сдачей решений в автоматизированную систему проверки.

Фонд оценочных средств дисциплины включает в себя задания для лабораторных работ, задания для самостоятельной работы, задания для контрольных работ, контрольные вопросы, вопросы к промежуточной аттестации. В рамках самостоятельной работы должны решаться задания, которые студент не успел решить в рамках аудиторных часов, выделенных под лабораторные занятия.

Фонд оценочных средств дисциплины приведён в приложении 1.

7. Данные для учета успеваемости студентов в БАРС

Таблица 1.1 – Таблица максимальных баллов по видам учебной деятельности

1	2	3	4	5	6	7	8	9
Семестр	Лекции	Лабораторные занятия	Практические занятия	Самостоятельная работа	Автоматизированное тестирование	Другие виды учебной деятельности	Промежуточная аттестация	Итого
5	10	30	0	20	0	0	40	100
6	10	30	0	20	0	0	40	100

Программа оценивания учебной деятельности студента

5 семестр

Лекции

Посещаемость, опрос, активность и др. за один семестр – от 0 до 10 баллов.

Лабораторные занятия

Контроль выполнения лабораторных заданий в течение одного семестра – от 0 до 30 баллов.

Практические занятия

Не предусмотрены.

Самостоятельная работа

Подготовка к лабораторным занятиям, контрольным работам, изучение материалов лекций, решение задач с использованием системы автоматической проверки задач в течении семестра – от 0 до 20 баллов.

Автоматизированное тестирование

Не предусмотрено.

Другие виды учебной деятельности

Не предусмотрено.

Промежуточная аттестация

от 0 до 40 баллов.

Промежуточная аттестация представляет собой зачет.

При проведении промежуточной аттестации

ответ на «отлично» оценивается от 31 до 40 баллов;

ответ на «хорошо» оценивается от 21 до 30 баллов;

ответ на «удовлетворительно» оценивается от 11 до 20 баллов;

ответ на «неудовлетворительно» оценивается от 0 до 10 баллов.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 5 семестр по дисциплине «Методы программирования» составляет 100 баллов.

Таблица 2.1 – Таблица пересчета полученной студентом суммы баллов по дисциплине «Методы программирования» в зачет

70 баллов и более	«зачтено» (при недифференцированной оценке)
меньше 70 баллов	«не зачтено»

Программа оценивания учебной деятельности студента

6 семестр

Лекции

Посещаемость, опрос, активность и др. за один семестр – от 0 до 10 баллов.

Лабораторные занятия

Контроль выполнения лабораторных заданий в течение одного семестра – от 0 до 30 баллов.

Практические занятия

Не предусмотрено.

Самостоятельная работа

Подготовка к лабораторным занятиям, контрольным работам, изучение материалов лекций, решение задач с использованием системы автоматической проверки задач в течении семестра – от 0 до 20 баллов.

Автоматизированное тестирование

Не предусмотрено.

Другие виды учебной деятельности

Не предусмотрено.

Промежуточная аттестация

от 0 до 40 баллов.

Промежуточная аттестация представляет собой экзамен .

При проведении промежуточной аттестации

ответ на «отлично» оценивается от 31 до 40 баллов;

ответ на «хорошо» оценивается от 21 до 30 баллов;

ответ на «удовлетворительно» оценивается от 11 до 20 баллов;

ответ на «неудовлетворительно» оценивается от 0 до 10 баллов.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 6 семестр по дисциплине «Методы программирования» составляет 100 баллов.

Таблица 2.2 – Таблица пересчета полученной студентом суммы баллов по дисциплине «Методы программирования» в оценку

91-100 баллов	«отлично»
81-90 баллов	«хорошо»
71-80 баллов	«удовлетворительно»
0-70 баллов	«неудовлетворительно»

8. Учебно-методическое и информационное обеспечение дисциплины

а) основная литература:

1. Круз, Роберт Л. Структуры данных и проектирование программ [Текст] = ✓✓
Data Structures and Program Design : [учеб. пособие] / Р. Л. Круз ; пер. 3-го
англ. изд. К. Г. Финогенова. - Москва : БИНОМ. Лаб. знаний, 2008
2. Васильев А. Java. Объектно-ориентированное программирование. Учебное ✓
пособие. Стандарт третьего поколения [Электронный ресурс] / А. Васильев. -
Санкт-Петербург : Питер, 2011. Режим доступа
<http://ibooks.ru/reading.php?short=1&isbn=978-5-49807-948-6>

б) дополнительная литература:

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы. Построение и ✓
анализ. - М.: Вильямс, 2005.
2. Кузнецов, В.А. Оптимизация на графах (алгоритмы и реализация) [Текст] : ✓
учеб. пособие / В. А. Кузнецов, А. М. Караваев ; Гос. образоват. учреждение
высш. проф. образования Петрозав. гос. ун-т. - Петрозаводск : Изд-во
ПетрГУ, 2007. - 183, [1] с. : рис. - Библиогр.: с. 182-183 (24 назв.). - ISBN 978-
5-8021-0771-3
3. Комбинаторные алгоритмы: множества, графы, коды/Быкова В.В. - ✓
Краснояр.: СФУ, 2015. - 152 с.: ISBN 978-5-7638-3155-9. Режим доступа
<http://znanium.com/bookread2.php?book=550333>

в) программное обеспечение и Интернет-ресурсы

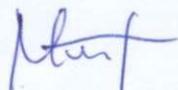
Стандартное программное обеспечение компьютерного класса, сборники
задач с поддержкой автоматизированного тестирования.

9. Материально-техническое обеспечение дисциплины

Лекционная аудитория с возможностью демонстрации электронных
презентаций при уровне освещения, достаточном для работы с конспектом.

Программа составлена в соответствии с требованиями ФГОС ВО по специальности 10.05.01 Компьютерная безопасность и специализации «Математические методы защиты информации» (квалификация «Специалист по защите информации»).

Автор
Старший преподаватель кафедры
математических основ информатики
и олимпиадного программирования



М.Р. Мирзаянов

Программа разработана в 2012 г. (одобрена на заседании кафедры теоретических основ компьютерной безопасности и криптографии от «25» мая 2012 года, протокол № 18).

Программа актуализирована в 2017 г. (одобрена на заседании кафедры математических основ информатики и олимпиадного программирования от «16» января 2017 года, протокол № 6).

Зав. кафедрой
математических основ информатики
и олимпиадного программирования
профессор



М.Б.Абросимов

Декан факультета
компьютерных наук
и информационных технологий
доцент



А.Г.Федорова