

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Рябинин Алексей Валерьевич
Должность: Ректор
Дата подписания: 18.08.2026 12:30:04
Уникальный программный ключ:
f5b92585d87b316237a7e4fb462e752b9baf0402

**Автономная некоммерческая организация высшего образования
«ИНСТИТУТ ЭКОНОМИКИ И УПРАВЛЕНИЯ В
ПРОМЫШЛЕННОСТИ»**

Лаборатория компьютерной безопасности



УТВЕРЖДЕНО
На заседании Ученого Совета АНО
ВО «Институт экономики и
управления в промышленности»
15 июня 2026 г. протокол № 7

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Методы программирования»

10.05.01 «Компьютерная безопасность»

Специализация:

Безопасность компьютерных систем и сетей

Квалификация выпускника:

специалист по защите информации

Форма обучения: *очная*

Москва, 2026 г.

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цели дисциплины - изучение основных методов и алгоритмов программирования, их применение для решения прикладных задач, связанных с поиском и сортировкой информации, формирование навыков к анализу эффективности функционирования алгоритмов.

1.2. Задачи освоения дисциплины

- изучить исходные понятия и формализации в языках программирования;
- освоить процессы представления, анализа работы языков программирования;
- сформировать умения использовать нормативную, правовую, справочную документацию и специальную литературу.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП

Дисциплина «Методы программирования» относится к дисциплинам блока Б1.

3. ПЕРЕЧЕНЬ ПЛАНИРУЕМЫХ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ

Процесс изучения дисциплины «Методы программирования» направлен на формирование следующих компетенций:

ОПК-7 - способен создавать программы на языке высокого уровня, применять существующие реализации структур данных и алгоритмов.

Компетенция	Результаты обучения, характеризующие сформированность компетенции
ОПК-7	Знать: - основные алгоритмы сортировки и поиска данных, комбинаторные и теоретико-графовые алгоритмы; - методы сортировок массивов данных различной длины и типов; - алгоритмы эффективного поиска информации в массивах и строках данных.
	Уметь: - разрабатывать и реализовывать на языке высокого уровня алгоритмы решения типовых профессиональных задач; - производить анализ и выбор эффективного алгоритма поиска и сортировки информации в

	различных массивах и строках данных с переменными любого типа; - оценивать технические возможности и вырабатывать рекомендации по разработке программного кода для эффективного выполнения проекта.
	Владеть: - навыками проектирования блок-схем, функциональных схем и алгоритмов реализации программного кода; - основами построения блок-схем, алгоритмов выполнения программ с выбранным методом сортировки или поиска информации; - методами оценки эффективности алгоритмов сортировки или поиска информации в различных массивах и строках данных с переменными любого типа

4. ОБЪЕМ ДИСЦИПЛИНЫ

Общая трудоемкость дисциплины «Методы программирования» составляет 7 з.е.

Распределение трудоемкости дисциплины по видам занятий

Виды учебной работы	Всего часов	Семестры	
		5	6
Контактная работа по видам занятий (всего)	144	72	72
В том числе:			
Лекции	72	36	36
Лабораторные работы (ЛР)	72	36	36
Самостоятельная работа	36	18	18
Курсовой проект		-	+
Часы на контроль	72	36	36
Виды промежуточной аттестации - экзамен		+	+
Общая трудоемкость: академические часы зач.ед.	252 7	126 3,5	126 3,5

5. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

5.1 Содержание разделов дисциплины и распределение трудоемкости по видам занятий

№ п/п	Наименование темы	Содержание раздела	Лекц	Лабор. работа	СРС	Всего, час
1	Понятие сложности алгоритма. Элементарные структуры данных.	Пространственная и временная сложность. Классы сложности. О-сложность алгоритмов. Зависимость сложности алгоритма и сложности данных. Определение сложности в О-обозначениях. Массивы. Список. Односвязный список. Двусвязный список. Кольцевой список	8	8	4	8
2	Абстрактные типы данных. Рекурсия и деревья	Очередь, стек. Очередь FIFO, очередь LIFO. Рекурсивные алгоритмы, функция вычисления факториала, сомнительная рекурсивная программа. Алгоритм Евклида, рекурсивная программа для оценки префиксных выражений. Деревья, обход дерева. Деревья с корнем, бинарные деревья.	8	8	4	20
3	Элементарные методы сортировки массивов данных	Сортировка выбором, сортировка вставками, пузырьковая сортировка. Характеристики производительности элементарных методов сортировки. Сортировка методом Шелла, быстрая сортировка. Характеристики производительности быстрой сортировки. Сортировка слиянием. Двухпутевое слияние.	8	8	4	12
4	Элементарные методы поиска информации в массивах и строках данных	Поиск с использованием индексации по ключам. Таблица символов, основывающаяся на индексированном по ключам массиве. Последовательный поиск. Таблица символов с использованием массива. Таблица символов с использованием связного списка. Бинарный поиск в таблице символов, основанной на массиве.	8	8	4	20
5	Бинарные деревья поиска	Таблица символов на базе дерева бинарного поиска. Создание дерева бинарного поиска. Сортировка с помощью BST-дерева. Вставка в BST – дерево (нерекурсивная). Производительность деревьев поиска. Поиск произвольного ключа в дереве. Необходимость балансировки деревьев. Рандомизированные BST-деревья. Балансировка BST-дерева. Удаление в рандомизированном BST-дереве.	8	8	4	20
6	Красно-черные деревья	Свойства красно-черных деревьев. Пример красно-черного дерева. Повороты. Вставка. Вставка узла в красно-черное дерево с n узлами. Удаление узла красно-черного дерева.	8	8	4	12
7	Точный поиск подстроки в строке	Задача поиска подстрок. Алгоритмы поиска подстрок. Время предварительной обработки и сравнения алгоритмов поиска. Простейший алгоритм поиска подстрок. Алгоритм Рабина-Карпа.	8	8	4	20

8	Алгоритмы на графах	Свойства и типы графов. Глоссарий. Полные графы. Двухдольный граф. АТД графа. Интерфейс АТД графа. Пример клиентской функции обработки графов. Клиентская функция печати графа. Интерфейс ввода/вывода для функций обработки графов. Пример клиентской программы обработки графов. Алгоритмы обхода графа в глубину. Алгоритм обхода графа в ширину. Алгоритм Дейкстры. Алгоритм Флойда. Практическая реализация алгоритма Флойда.	8	8	4	20
9	Алгоритмы нахождения минимального остовного дерева в графе.	Алгоритм Дейкстры-Прима. Принцип работы алгоритма Дейкстры-Прима. Алгоритм Крускала. Принцип работы алгоритма Крускала. Алгоритмы нахождения максимального потока транспортной сети. Остаточные сети. Увеличивающие пути. Разрезы транспортных сетей.	8	8	4	12
Итого			72	72	36	180

5.2 Перечень лабораторных занятий

1. Определение сложности алгоритмов – 4 ч.
2. Сортировка выбором – 4 ч.
3. Сортировка вставками – 4 ч.
4. Пузырьковая сортировка – 4 ч.
5. Сортировка методом Шелла – 4 ч.
6. Быстрая сортировка – 4 ч.
7. Последовательный поиск – 4 ч.
8. Бинарный поиска – 4 ч.
9. Алгоритм Робина-Карпа – 4 ч.

6. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ ПРОЕКТОВ (РАБОТ) И КОНТРОЛЬНЫХ РАБОТ

Учебным планом по дисциплине «Методы программирования» предусмотрено выполнение курсового проекта в 6-ом семестре. Примерные темы курсового проектирования:

- реализовать класс красно-черного дерева и проанализировать эффективность поиска в дереве;
- реализовать программу, использующую В-дерево для нахождения записей в файле по идентификатору. Оценить эффективность поиска записей;
- реализовать класс очереди с приоритетом и на его базе реализовать алгоритм пирамидальной сортировки;
- реализовать класс динамической хеш-таблицы, оценить эффективность поиска в ней по мере роста количества элементов;

- разработать программу, находящую максимальный пропускной поток транспортной системы, заданной графом, ребра которого имеют вес равный соответствующей пропускной способности отдельного участка.

Курсовой проект включает в себя решение практических задач из различных разделов курса «Методы программирования». Курсовой проект выполняется студентами в соответствии с заданным вариантом в соответствии с «Методическими указаниями к выполнению курсового проекта...».

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

7.1. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

7.1.1 Этап текущего контроля

Результаты текущего контроля знаний и межсессионной аттестации оцениваются по следующей системе:

«аттестован»;

«не аттестован».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Аттестован	Не аттестован
ОПК-7	Знать: - основные алгоритмы сортировки и поиска данных, комбинаторные и теоретико-графовые алгоритмы; - методы сортировок массивов данных различной длины и типов; - алгоритмы эффективного поиска информации в массивах и строках данных.	Активная работа на практических занятиях, отвечает на теоретические вопросы при защите курсового проекта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

Уметь: - разрабатывать и реализовывать на языке высокого уровня алгоритмы решения типовых профессиональных задач; - производить анализ и выбор эффективного алгоритма поиска и сортировки информации в различных массивах и строках данных с переменными любого типа; - оценивать технические возможности и вырабатывать рекомендации по разработке программного кода для эффективного выполнения проекта.	Решение стандартных практических задач, написание курсового проекта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах
Владеть: - навыками проектирования блок-схем, функциональных схем и алгоритмов реализации программного кода; - основами построения блок-схем, алгоритмов выполнения программ с выбранным методом сортировки или поиска информации; - методами оценки эффективности алгоритмов сортировки или поиска информации в различных массивах и строках данных с переменными любого типа	Решение прикладных задач в конкретной предметной области, выполнение плана работ по разработке курсового проекта	Выполнение работ в срок, предусмотренный в рабочих программах	Невыполнение работ в срок, предусмотренный в рабочих программах

7.1.2 Этап промежуточного контроля знаний

Результаты промежуточного контроля знаний оцениваются в 3 семестре по четырехбалльной системе:

- «отлично»;
- «хорошо»;
- «удовлетворительно»;
- «неудовлетворительно».

Компетенция	Результаты обучения, характеризующие сформированность компетенции	Критерии оценивания	Отлично	Хорошо	Удовл	Неудовл

ОПК-7	Знать: - основные алгоритмы сортировки и поиска данных, комбинаторные и теоретико-графовые алгоритмы; - методы сортировок массивов данных различной длины и типов; - алгоритмы эффективного поиска информации в массивах и строках данных	знание учебного материала и использование учебного материала в процессе выполнения заданий	Студент демонстрирует полное понимание учебного материала. Студент демонстрирует ярко выраженную способность использовать знания, умения, навыки в процессе выполнения заданий	Студент демонстрирует значительное понимание материала. Студент демонстрирует способность использовать знания, умения, навыки в процессе выполнения заданий	Студент демонстрирует частичное понимание материала. Способность студента продемонстрировать знание, умение, навык выражена слабо	1. Студент демонстрирует незначительное понимание материала. 2. Студент демонстрирует непонимание заданий. 3. У студента нет ответа. Не было попытки выполнить задания.
	Уметь: - разрабатывать и реализовывать на языке высокого уровня алгоритмы решения типовых профессиональных задач; - производить анализ и выбор эффективного алгоритма поиска и сортировки информации в различных массивах и строках данных с переменными любого типа; - оценивать технические возможности и вырабатывать рекомендации по разработке программного кода для эффективного выполнения проекта.	умение использовать учебный материал в процессе выполнения практических работ				
	Владеть: - навыками проектирования блок-схем, функциональных схем и алгоритмов реализации	применение учебного материала при решении практических задач				

программного кода; - основами построения блок- схем, алгоритмов выполнения программ с выбранным методом сортировки или поиска информации; - методами оценки эффективности алгоритмов сортировки или поиска информации в. различных массивах и строках данных с переменными любого типа						
--	--	--	--	--	--	--

7.2 Примерный перечень оценочных средств (типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности)

7.2.1 Примерный перечень заданий для подготовки к тестированию

1. Что **не является** определением алгоритма?

- 1) Точное предписание, однозначно определяющее вычислительный процесс, ведущий от варьируемых начальных данных к искомому результату
- 2) Процедура, которая принимает любой из возможных входных экземпляров задачи и преобразует его в соответствии с требованиями, указанными в условии задачи
- 3) Параметр, характеризующий данные, называют иногда объемом данных или сложностью данных
- 4) Конечный набор правил, однозначно раскрывающих содержание и последовательность выполнения операций для систематического решения определенного класса задач за конечное число шагов

2. Рассчитать сложность алгоритма $O(2*N+N^2)$

- 1) $O(N)$
- 2) $O(1)$
- 3) $O(2N)$
- 4) $O(N^2)$

3. Рассчитать НОД (алгоритм Евклида) 2 чисел – 357444 и 2154:

- 1) 1
- 2) 6
- 3) 2
- 4) 3

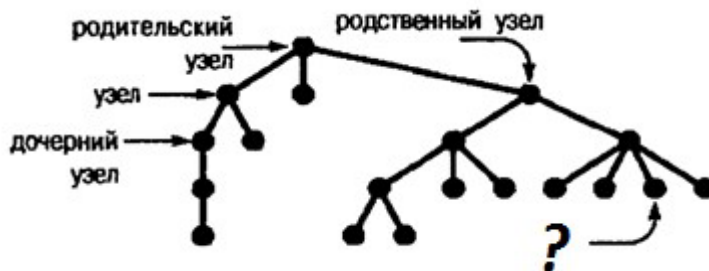
4. Начальный элемент односвязанного списка:

- 1) head list;
- 2) prev list;
- 3) next list;
- 4) nil

5. Стек, функционирующий по принципу "последним пришел – первым вышел":

- 1) FIFO,
- 2) LIFO
- 3) FIOS
- 4) FILO

6. Назовите элемент дерева, обозначенный знаком вопроса



- 1) корень
- 2) узел
- 3) лист
- 4) дочерний узел

7. Какое из выражений **не является** условием, чтобы граф считался деревом

- 1) Граф имеет N ребер и один цикл
- 2) Граф имеет N-1 ребер и является связным
- 3) Только один простой путь соединяет каждую пару вершин в графе
- 4) Граф является связным, но перестает быть таковым при удалении любого ребра

8. Определить тип сортировки массива

A S O R T I N G E X A M P L E
A S O R T I N G E X A M P L E
A A O R T I N G E X S M P L E
A A E R T I N G O X S M P L E
A A E E T I N G O X S M P L R
A A E E G I N T O X S M P L R
A A E E G I N T O X S M P L R
A A E E G I L T O X S M P L R
A A E E G I L M O X S T P N R
A A E E G I L M O X S T P N R
A A E E G I L M N X S T P O R
A A E E G I L M N O S T P X R
A A E E G I L M N O P T S X R
A A E E G I L M N O P R S X T
A A E E G I L M N O P R S X T
A A E E G I L M N O P R S T X
A A E E G I L M N O P R S T X

- 1) пузырьковая
- 2) вставками
- 3) выбором
- 4) шейкер

9. Вставьте пропущенные значения (вместо знаков вопроса)

Пузырьковая сортировка производит в среднем примерно? операций сравнения и? операций обмена как в среднем, так и в наихудшем случаях:

- 1) $N^2/2$, $N^2/2$
- 2) $N/2$, $N^2/2$
- 3) $N/2$, $N/2$
- 4) N^2 , N^2

10. Определить, каким видом сортировки является часть программы:

```
template <class Item>
```

```
void sort(Item a[] , int l, int r)
{
    if (r <= 1) return;
    int i = partition (a, l, r);
    sort(a, l, i-1);
    sort(a, i+1, r);
}
```

- 1) сортировка методом Шелла
- 2) быстрая сортировка
- 3) пузырьковая
- 4) вставками

7.2.4 Примерный перечень вопросов для подготовки к экзамену

1. Понятие сложности алгоритма. Пространственная и временная сложность.
2. Классы сложности. О-сложность алгоритмов. Зависимость сложности алгоритма и сложности данных.
3. Массивы. Список. Односвязный список. Двусвязный список. Кольцевой список.
4. Очередь, стек. Очередь FIFO, очередь LIFO.
5. Рекурсивные алгоритмы, функция вычисления факториала, сомнительная рекурсивная программа.
6. Алгоритм Евклида, рекурсивная программа для оценки префиксных выражений.
7. Деревья, обход дерева. Деревья с корнем, бинарные деревья.
8. Сортировка выбором, сортировка вставками, пузырьковая сортировка.
9. Характеристики производительности элементарных методов сортировки. Сортировка методом Шелла, быстрая сортировка.
10. Характеристики производительности быстрой сортировки. Сортировка слиянием. Двухпутевое слияние.
11. Поиск с использованием индексации по ключам. Таблица символов, основывающаяся на индексированном по ключам массиве. Последовательный поиск.
12. Таблица символов с использованием массива. Таблица символов с использованием связного списка. Бинарный поиск в таблице символов, основанной на массиве.
13. Таблица символов на базе дерева бинарного поиска. Создание дерева бинарного поиска. Сортировка с помощью BST-дерева.
14. Вставка в BST-дерево (нерекурсивная). Производительность деревьев поиска.
15. Поиск произвольного ключа в дереве. Необходимость балансировки деревьев.
16. Рандомизированные BST-деревья. Балансировка BST-дерева. Удаление в рандомизованном BST-дереве.
17. Свойства красно-черных деревьев. Пример красно-черного дерева.
18. Повороты. Вставка. Вставка узла в красно-черное дерево с n узлами.
19. Удаление узла красно-черного дерева.
20. Задача поиска подстрок. Алгоритмы поиска подстрок.
21. Время предварительной обработки и сравнения алгоритмов поиска.

- 22.Простейший алгоритм поиска подстрок. Алгоритм Рабина-Карпа.
- 23.Свойства и типы графов. Глоссарий. Полные графы.
- 24.Двухдольный граф. АДГ графа.
- 25.Интерфейс АДГ графа. Пример клиентской функции обработки графов.
- 26.Клиентская функция печати графа. Интерфейс ввода/вывода для функций обработки графов.
- 27.Пример клиентской программы обработки графов.
- 28.Алгоритмы обхода графа в глубину.
- 29.Алгоритм обхода графа в ширину.
- 30.Алгоритм Дейкстры.
- 31.Алгоритм Флойда. Практическая реализация алгоритма Флойда.
- 32.Алгоритм Дейкстры-Прима. Принцип работы алгоритма Дейкстры-Прима.
- 33.Алгоритм Крускала. Принцип работы алгоритма Крускала..
- 34.Алгоритмы нахождения максимального потока транспортной сети.
- 35.Остаточные сети. Увеличивающие пути. Разрезы транспортных сетей.

7.2.7 Паспорт оценочных материалов

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Понятие сложности алгоритма. Элементарные структуры данных	ОПК-7	Тест, решение практических задач, выполнение курсового проекта
2	Абстрактные типы данных. Рекурсия и деревья	ОПК-7	Тест, решение практических задач, выполнение курсового проекта
3	Элементарные методы сортировки массивов данных	ОПК-7	Тест, решение практических задач, выполнение курсового проекта
4	Элементарные методы поиска информации в массивах и строках данных	ОПК-7	Тест, решение практических задач, выполнение курсового проекта

5	Бинарные деревья поиска	ОПК-7	Тест, решение практических задач, выполнение курсового проекта
6	Красно-черные деревья	ОПК-7	Тест, решение практических задач, выполнение курсового проекта
7	Точный поиск подстроки в строке	ОПК-7	Тест, решение практических задач, выполнение курсового проекта
8	Алгоритмы на графах	ОПК-7	Тест, решение практических задач, выполнение курсового проекта
9	Алгоритмы нахождения минимального остовного дерева в графе.	ОПК-7	Тест, решение практических задач, выполнение курсового проекта

7.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

При преподавании дисциплины «Методы программирования» в качестве формы оценки знаний студентов используются: тесты, решение практических задач различной сложности, экзамен.

Тестирование осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных тест-заданий на бумажном носителе. Время тестирования 30 мин. Затем осуществляется проверка теста экзаменатором и выставляется оценка согласно методике выставления оценки, при проведении промежуточной аттестации.

Решение стандартных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методике выставления оценки, при проведении промежуточной аттестации.

Решение прикладных задач осуществляется, либо при помощи компьютерной системы тестирования, либо с использованием выданных задач на бумажном носителе. Время решения задач 30 мин. Затем осуществляется проверка решения задач экзаменатором и выставляется оценка, согласно методике выставления оценки, при проведении

промежуточной аттестации.

8 УЧЕБНО МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

8.1 Перечень учебной литературы, необходимой для освоения дисциплины

Основная:

1. Бульонков, М. А. Базовые понятия и методы программирования: учебное пособие / М. А. Бульонков, П. Г. Емельянов, И. Н. Скопин. — Новосибирск: Новосибирский государственный университет, 2023. — 366 с. — ISBN 978-5-4437-1495-0. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/134565.html>

2. Белик, А. Г. Алгоритмы и структуры данных: учебное пособие / А. Г. Белик, В. Н. Цыганенко. — Омск: Омский государственный технический университет, 2022. — 104 с. — ISBN 978-5-8149-3498-7. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/131186.html>

3. Белоусова, С. И. Основные принципы и концепции программирования на языке VBA в Excel: учебное пособие / С. И. Белоусова, И. А. Бессонова. — 5-е изд. — Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2024. — 191 с. — ISBN 978-5-4497-3312-2. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/142281.html>

Дополнительная:

1. Осипов, Д. Л. Методы программирования: лабораторный практикум / Д. Л. Осипов, А. С. Павлов. — Ставрополь: Северо-Кавказский федеральный университет, 2022. — 194 с. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/135707.html>

2. Васильковский, Д. В. Методы программирования микроконтроллеров серии AVR Mega. Лабораторный практикум: учебное пособие / Д. В. Васильковский, А. В. Руденко. — Москва: Национальный исследовательский ядерный университет «МИФИ», 2021. — 180 с. — ISBN 978-5-7262-2772-6. — Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. — URL: <https://www.iprbookshop.ru/125494.html>

3. Сергеев, М. Ю. Программирование задач с применением структурированных данных: лабораторный практикум / М. Ю. Сергеев, Н. И. Гребенникова, Т. И. Сергеева. — Воронеж: Воронежский государственный

технический университет, ЭБС АСВ, 2023. — 165 с. — ISBN 978-5-7731-1097-2.
— Текст: электронный // Цифровой образовательный ресурс IPR SMART:
[сайт]. — URL: <https://www.iprbookshop.ru/131026.html>

8.2 Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения, ресурсов информационно-телекоммуникационной сети «Интернет», современных профессиональных баз данных и информационных справочных систем:

<http://znanium.com/> (ЭБС Знаниум) <http://IPRbookshop.ru/>
(ЭБС IPRbooks (Айбукс)) <http://urait.ru/>
(Образовательная платформа «Юрайт»)

9. МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Специализированная лекционная аудитория, оснащенная оборудованием для лекционных демонстраций и проекционной аппаратурой.

10. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

По дисциплине «Методы программирования» читаются лекции, проводятся лабораторные работы, выполняется курсовой проект.

Основой изучения дисциплины являются лекции, на которых излагаются наиболее существенные и трудные вопросы, а также вопросы, не нашедшие отражения в учебной литературе.

На лабораторных работах проводится тестирование и решение задач в соответствии с темой занятия. Методики решения задач приведены в методических указаниях к практическим занятиям.

Большое значение по закреплению и совершенствованию знаний имеет самостоятельная работа студентов. Информацию о всех видах самостоятельной работы студенты получают на занятиях.

Контроль усвоения материала дисциплины производится проверкой курсового проекта, выполнения тестов, решения практических задач. Освоение дисциплины оценивается на экзамене.

Вид учебных занятий	Деятельность студента
---------------------	-----------------------

Лекция	Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначение вопросов, терминов, материала, которые вызывают трудности, поиск ответов в рекомендуемой литературе. Если самостоятельно не удастся разобраться в материале, необходимо сформулировать вопрос и задать преподавателю на лекции или на практическом занятии.
Лабораторная работа	Лабораторные работы позволяют научиться применять теоретические знания, полученные на лекции при решении конкретных задач. Чтобы наиболее рационально и полно использовать все возможности лабораторных работ для подготовки к ним необходимо: следует разобрать лекцию по соответствующей теме, ознакомиться с соответствующим разделом учебного пособия по данной дисциплине, проработать дополнительную литературу и источники, решить задачи для самостоятельного решения из соответствующего раздела методических указаний к лабораторным работам.
Самостоятельная работа	Самостоятельная работа студентов способствует глубокому усвоения учебного материала и развитию навыков самообразования. Самостоятельная работа предполагает следующие составляющие: - работа с текстами: учебниками, справочниками, дополнительной литературой, а также проработка конспектов лекций; - выполнение домашних заданий и расчетов; - работа над темами для самостоятельного изучения; - участие в работе студенческих научных конференций, олимпиад; - подготовка к промежуточной аттестации.
Подготовка к промежуточной аттестации	Готовиться к промежуточной аттестации следует систематически, в течение всего семестра. Интенсивная подготовка должна начинаться не позднее, чем за месяц-полтора до промежуточной аттестации. Данные перед экзаменом три дня эффективнее всего использовать для повторения и систематизации материала.