

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования «Новосибирский национальный исследовательский  
государственный университет» (Новосибирский государственный университет, НГУ)

**Факультет информационных технологий**

СОГЛАСОВАНО

Декан ФИТ НГУ

М.М. Лаврентьев

«03» июля 2019 г.



**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**Алгоритмы и структуры данных**

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА  
Направленность (профиль): Программная инженерия и компьютерные науки

Форма обучения: очная

Год обучения: 1, семестр: 1, 2

№	Вид деятельности	Семестр	
		1	2
1	Лекции, час.		
2	Практические занятия, час.	32	32
3	Лабораторные занятия, час.	32	32
4	Занятий в контактной форме без учета промежуточной аттестации, час, из них	64	64
5	в электронной форме, час.		
6	из них аудиторных занятий, час.	64	64
7	из них в активной и интерактивной форме, час.	64	64
8	консультаций, час.		
9	Самостоятельная работа, час.	8	8
10	в том числе на выполнение письменных работ, час		
11	Форма аттестации (экзамен, зачет, дифференцированный зачет), час	ДЗ	ДЗ
12	Всего зачетных единиц <sup>1</sup>	2	2

Новосибирск 2019

<sup>1</sup> С учетом выделенных часов на промежуточную аттестацию

Рабочая программа дисциплины составлена на основании федерального государственного образовательного стандарта (ФГОС) высшего образования - бакалавриат по направлению подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА.

Федеральный государственный образовательный стандарт (ФГОС) высшего образования - бакалавриат по направлению подготовки 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА введен в действие приказом Минобрнауки от 19.09.2017 № 929.

Место дисциплины в структуре учебного плана: Блок ФТД Факультативы, факультативная дисциплина.

Рабочая программа дисциплины утверждена решением Ученого совета факультета информационных технологий от 02.07.2019, протокол № 75.

Программу разработал:

доцент кафедры общей информатики ФИТ,  
кандидат физико-математических наук



Т.Г. Чурина

Заведующий кафедрой систем информатики ФИТ,  
доктор физико-математических наук



М.М. Лаврентьев

Ответственный за образовательную программу:

доцент кафедры систем информатики ФИТ,  
кандидат технических наук



А.А. Романенко

## Аннотация к рабочей программе дисциплины

«Алгоритмы и структуры данных»

Дисциплина «Алгоритмы и структуры данных» реализуется в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА, направленность (профиль): ПРОГРАММНАЯ ИНЖЕНЕРИЯ И КОМПЬЮТЕРНЫЕ НАУКИ по очной форме обучения на русском языке.

**Место в образовательной программе:** Дисциплина «Алгоритмы и структуры данных» развивает знания, умения и навыки, сформированные у обучающихся по результатам изучения следующих дисциплин: «Программирование», «Информатика», «Математическая логика и теория алгоритмов», «Алгебра и геометрия», «Математический анализ».

Дисциплина «Алгоритмы и структуры данных» реализуется в 1, 2 семестрах в рамках Блока ФТД Факультативы и является факультативной дисциплиной.

Дисциплина «Алгоритмы и структуры данных» направлена на формирование компетенций:

Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений (УК-2), в части следующих индикаторов достижения компетенции:

УК-2.2 Уметь: проводить анализ поставленной цели и формулировать задачи, которые необходимо решить для ее достижения; анализировать альтернативные варианты решений для достижения намеченных результатов; использовать нормативно- правовую документацию в сфере профессиональной деятельности

Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности (ОПК-1), в части следующих индикаторов достижения компетенции:

ОПК-1.2 Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и обще-инженерных знаний, методов математического анализа и моделирования

Способен разрабатывать алгоритмы и программы, пригодные для практического применения (ОПК-8), в части следующих индикаторов достижения компетенции:

ОПК-8.1 Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения

ОПК-8.2 Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули

ОПК-8.3 Владеть: языком программирования; навыками отладки и тестирования работоспособности программы

Способен осваивать методики использования программных средств для решения практических задач (ОПК-9) в части следующих индикаторов достижения компетенции:

ОПК-9.2 Уметь: находить и анализировать техническую документацию по использованию программного средства, выбирать и использовать необходимые функции программных средств для решения конкретной задачи

Основной целью освоения дисциплины «Алгоритмы и структуры данных» является знакомство как с типовыми задачами программирования и основными моделями и методами их решения, на примере которых дается представление об искусстве программирования, так и с современными, постоянно развивающимися, используемыми в ведущих международных чемпионатах по информатике и программированию. Целью является также совершенствование владения языками программирования и техникой программирования. Обучение дисциплины ведется с использованием автоматической системы тестирования NSUts. Индивидуальные задания занесены в систему NSUts в виде туров. Лабораторные работы сопровождается итоговым анализом достигнутого, обсуждением эффективности выполненных задач, а также границ применимости обрабатываемых методов и приемов.

Изучение дисциплины опирается на материал курсов «Программирование», «Информатика», «Математическая логика и теория алгоритмов», «Алгебра и геометрия», «Математический анализ».

#### **Перечень основных разделов дисциплины:**

Способы реализации данных в ЭВМ.

Оценка сложности вычислительных алгоритмов и общие методы решения вычислительных задач.

Реализация алгоритмов поиска и сортировки. Хеширование.

Алгоритмические аспекты решения задач из теории чисел.

Геометрические задачи, эффективные способы решения.

Способы реализации списка, стека, очереди, задачи на их использование.

Абстрактные структуры данных: графы, деревья. Реализация задач с использованием графов и деревьев.

Кучи, красно-черное дерево, декартово дерево, дерево отрезков, дерево Фенвика, суффиксные дерево и массив, решение задач с их использованием.

решение задач с их использованием.

Система непересекающихся множеств (СНМ), способы реализации.

Решение задач с использованием полного перебора, алгоритмов с возвратом, метода ветвей и границ, динамического программирования.

Стратегии игр, разбор задач.

Задачи теории расписаний, решение задач.

Потоки в сетях, паросочетания их использование.

Преподавание дисциплины предусматривает следующие виды учебной работы: практические занятия, лабораторные занятия с использованием автоматизированной системы тестирования NSUts, самостоятельную работу. Самостоятельная работа включает: разбор материала, полученного во время практических занятий, реализацию алгоритмов и сдачу решений в систему NSUts.

Общая трудоемкость дисциплины составляет - 4 зачетных единицы (144 часа).

**Правила аттестации по дисциплине.** Для осуществления текущего контроля по дисциплине «Алгоритмы и структуры данных» планом дисциплины предусмотрено выполнение заданий портфолио:

текущих заданий в системе NSUts и сдача этих заданий с объяснением кода программ преподавателю.

Диф.зачет проводится по результатам оценивания портфолио студента.

Оценка «отлично» соответствует продвинутому уровню сформированности компетенции.

Оценка «хорошо» соответствует базовому уровню сформированности компетенции.

Оценка «удовлетворительно» соответствует пороговому уровню сформированности компетенции.

#### **Учебно-методическое обеспечение дисциплины.**

В преподавании дисциплины используются выполненные авторами презентации с демонстрацией классических и современных алгоритмов, кодом, реализующим изучаемые алгоритмы, и разбором задач.

В преподавании дисциплины используются изданные авторами учебные пособия:

1. Чурина Т.Г., Нестеренко Т.В. Методы программирования: алгоритмы и структуры данных. Часть 3. Динамические структуры данных, алгоритмы на графах: учеб. Пособие/ Новосиб. гос. ун-т.– Новосибирск: РИЦ НГУ, 2014. ISBN 978-5-4437-0278-0, 216 с. (60 экз)

А также используются:

2. Система тестирования NSUts [Электронный ресурс] URL: <https://olympic.nsu.ru/nsuts-new/login.cgi>

## 1. Внешние требования к дисциплине

Таблица 1.1

<b>Компетенция УК-2</b> Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений, в части следующих индикаторов достижения компетенции:	
УК-2.2	Уметь: проводить анализ поставленной цели и формулировать задачи, которые необходимо решить для ее достижения; анализировать альтернативные варианты решений для достижения намеченных результатов; использовать нормативно- правовую документацию в сфере профессиональной деятельности
<b>Компетенция ОПК-1</b> Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности, в части следующих индикаторов достижения компетенции:	
ОПК-1.2	Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и обще-инженерных знаний, методов математического анализа и моделирования
<b>Компетенция ОПК-8</b> Способен разрабатывать алгоритмы и программы, пригодные для практического применения, в части следующих индикаторов достижения компетенции:	
ОПК-8.1	Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения
ОПК-8.2	Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы , интегрировать программные модули
ОПК-8.3	Владеть: языком программирования; навыками отладки и тестирования работоспособности программы
<b>Компетенция ОПК-9</b> Способен осваивать методики использования программных средств для решения практических задач, в части следующих индикаторов достижения компетенции:	
ОПК-9.2	Уметь: находить и анализировать техническую документацию по использованию программного средства, выбирать и использовать необходимые функции программных средств для решения конкретной задачи

## 2. Требования к результатам освоения дисциплины

Таблица 2.1

Результаты изучения дисциплины по уровням освоения (иметь представление, знать, уметь, владеть)	Формы организации занятий		
	Практические занятия	Лабораторные работы	Самостоятельная работа
<b>УК-2.2</b> Уметь: проводить анализ поставленной цели и формулировать задачи, которые необходимо решить для ее достижения; анализировать альтернативные варианты решений для достижения намеченных результатов; использовать нормативно- правовую документацию в сфере профессиональной деятельности			
1. Уметь обобщать, анализировать и строить математическую модель задачи	+	+	+
2. Уметь анализировать алгоритм на быстродействие его выполнения, определять класс сложности алгоритма	+	+	+

<b>ОПК-1.2</b> Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и обще-инженерных знаний, методов математического анализа и моделирования			
3. Уметь использовать стандартные методы и схемы решений задач	+	+	+
4. Уметь эффективно реализовать решение задачи на основе построенной модели	+	+	+
<b>ОПК-8.1</b> Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения			
5. Знать основные структуры данных и способы их реализации на языке высокого уровня	+	+	+
6. Знать общие методы решения вычислительных задач	+	+	+
7. Знать классические алгоритмы и способы их реализации в соответствующей среде разработки	+	+	+
<b>ОПК-8.2</b> Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули			
8. Уметь применять типовые методы для решения задач программирования повышенной сложности	+	+	+
9. Уметь тестировать программы и оценивать их производительность и эффективность	+	+	+
<b>ОПК-8.3</b> Владеть: языком программирования; навыками отладки и тестирования работоспособности программы			
10. Владеть использованием стандартных методов и схем решений задач	+	+	+
11. Владеть эффективными способами реализации современных алгоритмов	+	+	+
<b>ОПК-9.2</b> Уметь: находить и анализировать техническую документацию по использованию программного средства, выбирать и использовать необходимые функции программных средств для решения конкретной задачи			
12. Владеть навыками работы в системе тестирования NSUts	+	+	+

### 3. Содержание и структура учебной дисциплины

Таблица 3.2

Темы семинарских занятий	Активные формы, часы	Часы	Ссылки на результаты обучения	
<b>Семестр 1</b>				
1. Способы реализации данных в ЭВМ Алгоритмы перевода целых чисел из одной в-с.с. в другую. Модификации	4	4	3, 5, 7, 12	Разбирают алгоритмы, реализуют их на C/C++, решают задачи по разделам,

<p>этих алгоритмов для перевода действительных чисел.</p> <p>Дополнительный код, перенос и переполнение.</p> <p>Длинная арифметика, способы реализации, маски, решение задач с их использованием</p> <p>Элементы теорий вероятностей, информации и кодирования.</p>				описанным в Теме 1.
<p><u>2. Оценка сложности вычислительных алгоритмов и общие методы решения вычислительных задач</u></p> <p>Размер задачи как характеристика объема входных данных. Временная и емкостная сложность программы. Классы эффективности алгоритмов. Основные методы программирования: цикл, ветвление, рекурсия. Рекурсивный переход, правила выхода, ветвящаяся и хвостовая рекурсия.</p>	2	2	2, 6	Разбирают теорию по разделу Темы 2.
<p><u>3. Реализация алгоритмов поиска и сортировки. Хеширование</u></p> <p>Методы простого поиска в массиве, методы поиска подстроки в строке: алгоритм прямого перебора, алгоритм Бойера—Мура, Рабина-Карпа, Кнута-Морриса-Пратта, способы реализации, оценки сложности.</p> <p>Перестановки набора. Алгоритмы перебора перестановок, реализация.</p> <p>Методы сортировки массива: шейкер-сортировка; метод Шелла.</p>	8	8	1, 3, 5, 7, 10	Разбирают алгоритмы и способы их реализации по разделам, описанным в Теме 3.

Быстрая сортировка Хоара; сортировка деревом, пирамидальная сортировка; реализация с использованием кучи. Хэш-функции, расширяемое хеширование.				
<u>4. Алгоритмические аспекты решения задач из теории чисел</u> Операции по модулю, быстрое возведение в степень, вычисление полинома через схему Горнера, НОД и НОК, алгоритм Евклида, двоичный алгоритм, обратное по модулю m, мультипликативная группа вычетов, простые числа в разложении на множители, системы сравнений по модулю, диофантовы уравнения, рациональная арифметика, способы реализации	4	4	3, 6	Разбирают теорию и алгоритмы, реализуют их на C/C++, решают задачи по разделам, описанным в Теме 4.
<u>5. Геометрические задачи, эффективные способы решения</u> Векторная арифметика, определение взаимного расположения точки и прямой, двух отрезков, точки и многоугольника, точки и плоскости, задачи с участием окружностей, отрезков и дуг, определение вида и площади многоугольника. Построение выпуклой оболочки, алгоритм Грэхема. Задачи на обработку событий, метод движущейся прямой / вращающегося луча.	6	6	1, 10, 11	Разбирают теорию и алгоритмы, реализуют их на C/C++, решают задачи по разделам, описанным в Теме 5.
<u>6. Способы реализации списка, стека, очереди, задачи с их использованием</u> Разновидности списков:	4	4	5, 12	Разбирают способы реализации структур данных,

одно/двусвязные; циклические; иерархические. Способы реализации списков (статика, динамика). Алгоритмы поиска и включения для списков, анализ их эффективности. Реализация стека, задача о преобразовании инфиксной формы записи выражения в постфиксную и вычисление значения полученного выражения. Реализация очереди, Основные операции, способы реализации на различных базовых представлениях. Решение задачи о лабиринтах.				решают задачи по разделам, описанным в Теме 6.
<u>7. Абстрактные структуры данных: графы, деревья. Способы реализации</u> Графы, пути и маршруты, модели представления в ЭВМ. Дерево, как частный вид графа. Дерево двоичного поиска. Уравновешанное дерево. Алгоритмы включения и удаления. Левое/правое скобочное представление деревьев. Обход вершин графа методом поиска в глубину, реализация. Свойства поиска в глубину, использование стека. Обход графа методом в ширину. Реализация с помощью очереди.	4	4	5, 7, 12	Разбирают теорию и алгоритмы, реализуют их на C/C++, решают задачи по разделам, описанным в Теме 7.
<b>Итого</b>	<b>32</b>	<b>32</b>		
<b>Семестр 2</b>				
<u>8. Классические задачи на графах и деревьях. Реализация задач с использованием графов и деревьев.</u> Реализация алгоритма поиска	4	4	2, 5, 7	Разбирают алгоритмы и способы их реализации по разделам,

<p>двусвязных компонент и точек сочленения графа.</p> <p>Кратчайшие пути в лабиринте.</p> <p>Топологическая сортировка, реализация на иерархических списках.</p> <p>Поиск кратчайших путей в графе, алгоритмы Дейкстры, Беллмана-Форда, Флойда-Уоршелла.</p> <p>Алгоритм Флойда. Эффективная реализация алгоритмов в том числе на куче, оценки сложности.</p>				описанным в Теме 8.
<p><u>9. Кучи, красно-черное дерево, декартово дерево, дерево отрезков, дерево Фенвика, суффиксное дерево и массив, решение задач с их использованием</u></p> <p>Бинарные кучи, Фибоначчиевы кучи, реализация. Алгоритм построения суффиксного массива, его использование в алгоритмах поиска.</p> <p>Создание дерева отрезков, запрос суммы на отрезке, запрос обновления. Дерево Фенвика для суммы, для максимума</p>	8	8	1, 11	Разбирают теорию, алгоритмы, эффективные способы реализации, пишут программы на C/C++, решают задачи по разделам, описанным в Теме 9.
<p><u>10. Система непересекающихся множеств (СНМ), способы реализации</u></p> <p>Каркас графа минимальной стоимости, алгоритмы Краскала, Прима, Прима-Краскала. Система непересекающихся множеств, эвристика объединением по рангу, эвристика сжатия путей, реализация, оценки сложности.</p>	4	4	1, 4, 9	Разбирают теорию, алгоритмы, эффективные способы реализации алгоритмов по Теме 10.
<p><u>11. Решение задач с использованием</u></p>	4	4	2, 3, 7, 12	Разбирают

<p><u>полного перебора, алгоритмов с возвратом, методов ветвей и границ, динамического программирования</u></p> <p>Задача коммивояжера, гамильтоновы циклы, алгоритмы Флери, алгоритм об устойчивых браках, способы реализации.</p> <p>Алгоритмы с возвратом: расстановка ферзей, обход шахматной доски конем.</p> <p>Динамическое программирование как решение задач с помощью табличной техники. Алгоритм Ахо о преобразовании строк, задачи о наибольшей общей подпоследовательности, о делимости, об умножении матриц, о расстановке скобок в выражении, о раскрое многоугольника, об умножении матриц.</p> <p>Метод ветвей и границ, задачи о рюкзаке, реализация.</p>				<p>алгоритмы, реализуют их на C/C++, решают задачи по разделам, описанным в Теме 11.</p>
<p><u>Раздел 12. Стратегии игр, разбор задач</u></p> <p>Определение игры, определение оптимальной стратегии, поиск оптимальных стратегий.</p> <p>Определение неподвижной точки.</p>	4	4	1, 2, 11	<p>Разбирают теорию, примеры задач, реализуют игру по Теме 12.</p>
<p><u>Раздел 13. Задачи теории расписаний, решение задач</u></p> <p>Обзор различных направлений исследований в теории расписаний: оптимальное планирование, календарное планирование, задачи классов Flow-, Open-, Job-Shop.</p>	4	4	2, 3, 6, 11	<p>Разбирают теорию, реализуют задачи по Теме 12.</p>

<p><u>Раздел 14. Потоки в сетях, паросочетания их использование</u></p> <p>Потоки в сетях, понятие остаточной сети. Теорема о максимальном потоке и минимальном разрезе.</p> <p>Метод Форда-Фалкерсона и алгоритм Эдмунда-Карпа. Алгоритм поиска максимального потока методом проталкивания предпотока.</p> <p>Паросочетания, поиск максимальных паросочетаний в двудольном графе.</p>	4	4	4, 7, 8, 12	Разбирают теорию, реализуют алгоритмы, решают задачи по Теме 13.
<b>Итого</b>	<b>32</b>	<b>32</b>		

Темы лабораторных занятий	Активные формы, часы	Часы	Ссылки на результаты обучения	
<b>Семестр 1</b>				
<p><u>1. Способы реализации данных в ЭВМ</u></p> <p>Алгоритмы перевода целых чисел из одной в-с.с. в другую. Модификации этих алгоритмов для перевода действительных чисел.</p> <p>Дополнительный код, перенос и переполнение.</p> <p>Длинная арифметика, способы реализации, маски, решение задач с их использованием</p> <p>Элементы теорий вероятностей, информации и кодирования.</p>	4	4	3, 5, 7, 12	Разбирают алгоритмы, реализуют их на С/С++, решают задачи по разделам, описанным в Теме 1.
<p><u>2. Оценка сложности вычислительных алгоритмов и общие методы решения вычислительных задач</u></p> <p>Размер задачи как характеристика объема входных данных. Временная</p>	2	2	2, 6	Разбирают теорию по разделу Темы 2.

<p>и емкостная сложность программы.</p> <p>Классы эффективности алгоритмов.</p> <p>Основные методы программирования: цикл, ветвление, рекурсия. Рекурсивный переход, правила выхода, ветвящаяся и хвостовая рекурсия.</p>				
<p><u>3. Реализация алгоритмов поиска и сортировки. Хеширование</u></p> <p>Методы простого поиска в массиве, методы поиска подстроки в строке: алгоритм прямого перебора, алгоритм Бойера—Мура, Рабина-Карпа, Кнута-Морриса-Пратта, способы реализации, оценки сложности.</p> <p>Перестановки набора. Алгоритмы перебора перестановок, реализация.</p> <p>Методы сортировки массива: шейкер-сортировка; метод Шелла.</p> <p>Быстрая сортировка Хоара; сортировка деревом, пирамидальная сортировка; реализация с использованием кучи. Хэш-функции, расширяемое хеширование.</p>	8	8	1, 3,5, 7, 10	<p>Разбирают алгоритмы и способы их реализации по разделам, описанным в Теме 3.</p>
<p><u>4. Алгоритмические аспекты решения задач из теории чисел</u></p> <p>Операции по модулю, быстрое возведение в степень, вычисление полинома через схему Горнера, НОД и НОК, алгоритм Евклида, двоичный алгоритм, обратное по модулю <math>m</math>, мультипликативная группа вычетов, простые числа в разложении на множители, системы сравнений по</p>	4	4	3, 6	<p>Разбирают теорию и алгоритмы, реализуют их на C/C++, решают задачи по разделам, описанным в Теме 4.</p>

модулю, диофантовы уравнения, рациональная арифметика, способы реализации				
<p><u>5. Геометрические задачи, эффективные способы решения</u></p> <p>Векторная арифметика, определение взаимного расположения точки и прямой, двух отрезков, точки и многоугольника, точки и плоскости, задачи с участием окружностей, отрезков и дуг, определение вида и площади многоугольника.</p> <p>Построение выпуклой оболочки, алгоритм Грэхема.</p> <p>Задачи на обработку событий, метод движущейся прямой / вращающегося луча.</p>	6	6	1, 10, 11	Разбирают теорию и алгоритмы, реализуют их на C/C++, решают задачи по разделам, описанным в Теме 5.
<p><u>6. Способы реализации списка, стека, очереди, задачи с их использованием</u></p> <p>Разновидности списков: одно/двусвязные; циклические; иерархические. Способы реализации списков (статика, динамика).</p> <p>Алгоритмы поиска и включения для списков, анализ их эффективности.</p> <p>Реализация стека, задача о преобразовании инфиксной формы записи выражения в постфиксную и вычисление значения полученного выражения.</p> <p>Реализация очереди, Основные операции, способы реализации на различных базовых представлениях.</p> <p>Решение задачи о лабиринтах.</p>	4	4	5, 12	Разбирают способы реализации структур данных, решают задачи по разделам, описанным в Теме 6.

<p><u>7. Абстрактные структуры данных: графы, деревья. Способы реализации</u></p> <p>Графы, пути и маршруты, модели представления в ЭВМ. Дерево, как частный вид графа. Дерево двоичного поиска. Уравновешанное дерево. Алгоритмы включения и удаления. Левое/правое скобочное представление деревьев.</p> <p>Обход вершин графа методом поиска в глубину, реализация. Свойства поиска в глубину, использование стека.</p> <p>Обход графа методом в ширину. Реализация с помощью очереди.</p>	4	4	5, 7, 12	Разбирают теорию и алгоритмы, реализуют их на C/C++, решают задачи по разделам, описанным в Теме 7.
<b>Итого</b>	<b>32</b>	<b>32</b>		
<b>Семестр 2</b>				
<p><u>8. Классические задачи на графах и деревьях. Реализация задач с использованием графов и деревьев.</u></p> <p>Реализация алгоритма поиска двусвязных компонент и точек сочленения графа.</p> <p>Кратчайшие пути в лабиринте.</p> <p>Топологическая сортировка, реализация на иерархических списках.</p> <p>Поиск кратчайших путей в графе, алгоритмы Дейкстры, Беллмана-Форда, Флойда-Уоршелла.</p> <p>Алгоритм Флойда. Эффективная реализация алгоритмов в том числе на куче, оценки сложности.</p>	4	4	2, 5, 7	Разбирают алгоритмы и способы их реализации по разделам, описанным в Теме 8.
<p><u>9. Кучи, красно-черное дерево, декартово дерево, дерево отрезков,</u></p>	8	8	1, 11	Разбирают теорию,

<p><u>дерево Фенвика, суффиксные дерево и массив, решение задач с их использованием</u></p> <p>Бинарные кучи, Фибоначчиевы кучи, реализация. Алгоритм построения суффиксного массива, его использование в алгоритмах поиска. Создание дерева отрезков, запрос суммы на отрезке, запрос обновления. Дерево Фенвика для суммы, для максимума</p>				<p>алгоритмы, эффективные способы реализации, пишут программы на С/С++, решают задачи по разделам, описанным в Теме 9.</p>
<p><u>10. Система непересекающихся множеств (СНМ), способы реализации</u></p> <p>Каркас графа минимальной стоимости, алгоритмы Краскала, Прима, Прима-Краскала. Система непересекающихся множеств, эвристика объединением по рангу, эвристика сжатия путей, реализация, оценки сложности.</p>	4	4	1	<p>Разбирают теорию, алгоритмы, эффективные способы реализации алгоритмов по Теме 10.</p>
<p><u>11. Решение задач с использованием полного перебора, алгоритмов с возвратом, методов ветвей и границ, динамического программирования</u></p> <p>Задача коммивояжера, гамильтоновы циклы, алгоритмы Флери, алгоритм об устойчивых браках, способы реализации.</p> <p>Алгоритмы с возвратом: расстановка ферзей, обход шахматной доски конем.</p> <p>Динамическое программирование как решение задач с помощью табличной техники. Алгоритм Ахо о преобразовании строк, задачи о</p>	4	4	2, 3, 7, 12	<p>Разбирают алгоритмы, реализуют их на С/С++, решают задачи по разделам, описанным в Теме 11.</p>

<p>наибольшей общей подпоследовательности, о делимости, об умножении матриц, о расстановке скобок в выражении, о раскрое многоугольника, об умножении матриц.</p> <p>Метод ветвей и границ, задачи о рюкзаке, реализация.</p>				
<p><u>Раздел 12. Стратегии игр, разбор задач</u></p> <p>Определение игры, определение оптимальной стратегии, поиск оптимальных стратегий.</p> <p>Определение неподвижной точки.</p>	4	4	1, 2, 11	Разбирают теорию, примеры задач, реализуют игру по Теме 12.
<p><u>Раздел 13. Задачи теории расписаний, решение задач</u></p> <p>Обзор различных направлений исследований в теории расписаний: оптимальное планирование, календарное планирование, задачи классов Flow-, Open-, Job-Shop.</p>	4	4	2, 3, 6, 11	Разбирают теорию, реализуют задачи по Теме 12.
<p><u>Раздел 14. Потoki в сетях, паросочетания их использование</u></p> <p>Потоки в сетях, понятие остаточной сети. Теорема о максимальном потоке и минимальном разрезе.</p> <p>Метод Форда-Фалкерсона и алгоритм Эдмодса-Карпа. Алгоритм поиска максимального потока методом проталкивания предпотока.</p> <p>Паросочетания, поиск максимальных паросочетаний в двудольном графе.</p>	4	4	4, 7, 8, 9, 12	Разбирают теорию, реализуют алгоритмы, решают задачи по Теме 13.
<b>Итого</b>	<b>32</b>	<b>32</b>		

#### 4. Самостоятельная работа студентов

Таблица 4.1

№	Виды самостоятельной работы	Ссылки на результаты обучения	Часы на выполнение	Часы на консультации
<b>Семестр 1</b>				
1.	Самостоятельный разбор материала разделов 1-7	1, 2,3,4, 5, 6, 7	4	0
	Обучающиеся повторяют математические понятия и алгоритмы, используя рекомендованную литературу и презентации авторов программы с демонстрацией классических и современных алгоритмов, кодом, реализующим изучаемые алгоритмы, и разбором задач, расположенные по ссылке <a href="http://programming.iis.nsk.su/node/1759">http://programming.iis.nsk.su/node/1759</a>			
2.	Подготовка и сдача решений задач разделов 1, 3, 5, 6 в систему тестирования NSUts	8,9, 10,11, 12	4	0
	Обучающиеся решают практические задачи, входящие в портфолио.			
	Итого		8	0
<b>Семестр 2</b>				
1.	Самостоятельный разбор материала разделов 8-14	2, 5, 6, 7	4	0
	Обучающиеся повторяют математические понятия и алгоритмы, используя рекомендованную литературу и презентации авторов программы с демонстрацией классических и современных алгоритмов, кодом, реализующим изучаемые алгоритмы, и разбором задач, расположенные по ссылке <a href="http://programming.iis.nsk.su/node/1759">http://programming.iis.nsk.su/node/1759</a>			
2.	Подготовка и сдача решений задач разделов 8-10,14 в систему тестирования NSUts	9, 10, 11, 12	4	0
	Обучающиеся решают практические задачи, входящие в портфолио.			
	Итого		8	0

При сдаче заданий устно и в систему NSUts необходимо следовать методическим указаниям и требованиям, приведенными ниже.

#### Методические указания по выполнению лабораторных работ

- лабораторные работы выполняются на языке программирования C/C++;
- программа должна сопровождаться кратким описанием в виде комментария в заголовке программы, если она несложная, или отдельного текста, описывающего, что делает эта программа, какой алгоритм использует и как вызывается, с какими параметрами. Каждая функция должна предваряться комментарием с информацией о назначении функции, типах параметров и возвращаемого значения и с кратким описанием алгоритма, если он нетривиален.
- результатом выполнения лабораторной работы является набор файлов на языке программирования, если программа сдается устно;
- результатом выполнения лабораторной работы является один файл на языке программирования, если программа сдается в автоматизированной системе тестирования NSUts.

- исполняемая программа должна проходить все тесты, подготовленные преподавателем к лабораторной работе, если программа сдается устно.
- исполняемая программа должна проходить все тесты, загруженные в системе NSUts к лабораторной работе, если программа сдается в этой системе.

### Требования к сдаваемым программам

- Программа должна быть написана так, чтобы работа с ней не требовала вашего участия, в частности, должна быть предусмотрена выдача программой информации по ее назначению и использованию (способу вызова).
- Ввод должен быть по возможности организован так, чтобы программа могла получать входные данные с клавиатуры или из файла, имя которого подается программе параметром, а результат выдавать на монитор или в файл, имя которого также подается параметром.
- При вводе программа должна корректно реагировать на ошибочные входные данные. При обнаружении ошибок программа должна правильно их диагностировать и давать рекомендации по их устранению.
- Вывод результата работы должен быть исчерпывающим и ясным.
- Следует продумать систему тестов, проверяющих правильность работы программы.

## 5. Образовательные технологии

Используются традиционные технологии, активные и интерактивные формы проведения занятий. В практических занятиях используется видео, презентации, на которых представлена работа алгоритмов с использованием элементов анимации; обсуждаются программные системы с использованием видеопроектора.

Во время лабораторных занятий реализуются на языке C++ или Java изученные алгоритмы, устно сдаются задачи, поставленные преподавателем по темам курса, используется автоматическое компьютерное тестирование программ, задачи сдаются в автоматизированную систему тестирования NSUts.

В ходе реализации учебного процесса применяются следующие интерактивные формы организации учебных занятий (таблица 5.1).

Таблица 5.1

1	Технологии проблемного обучения	УК-2
<b>Формируемые умения:</b> 1. Уметь обобщать, анализировать и строить математическую модель задачи. 2. Уметь анализировать алгоритм на быстродействие его выполнения, определять класс сложности алгоритма		
<b>Краткое описание применения:</b> Постановка под руководством преподавателя проблемных задач и активная самостоятельная деятельность обучающихся по их разрешению, сопровождающаяся обсуждением результатов.		
2	Портфолио	ОПК-1
<b>Формируемые умения:</b> 3. Уметь использовать стандартные методы и схемы решений задач. 4. Уметь эффективно реализовать решение задачи на основе построенной модели.		
<b>Краткое описание применения:</b> студенты ведут портфолио в системе NSUts, которое является основой для проведения аттестации по дисциплине.		
3	Портфолио	ОПК-8, ОПК-9
<b>Формируемые умения:</b> 8. Уметь применять типовые методы для решения задач программирования повышенной сложности. 9. Уметь тестировать программы и оценивать их производительность и эффективность. 12. Владеть навыками работы в		

системе тестирования NSUts.
<b>Краткое описание применения:</b> Постановка под руководством преподавателя проблемных задач и активная самостоятельная деятельность обучающихся по их разрешению, сопровождающаяся обсуждением результатов. Студенты ведут портфолио (коллекцию работ) в системе NSUts, которое является основой для проведения аттестации по дисциплине.

Для организации и контроля самостоятельной работы студентов, а также проведения консультаций применяются информационно-коммуникационные технологии (таблица 5.2).

Таблица 5.2

Информирование	zaic101@gmail.com <a href="https://olympic.nsu.ru/">https://olympic.nsu.ru/</a> <a href="https://olympic.nsu.ru/nsuts-new/login.cgi">https://olympic.nsu.ru/nsuts-new/login.cgi</a>
Консультирование	<a href="https://olympic.nsu.ru/nsuts-new/login.cgi">https://olympic.nsu.ru/nsuts-new/login.cgi</a>
Контроль	tanch@iis.nsk.su
Размещение учебных материалов	<a href="http://programming.iis.nsk.su/node/1759">http://programming.iis.nsk.su/node/1759</a>

## 6. Правила аттестации студентов по учебной дисциплине

**Текущая аттестация** по дисциплине «Алгоритмы и структуры данных» осуществляется на практических занятиях и заключается в выполнении заданий, входящих в рамки портфолио:

- выполнение индивидуальных заданий и устная сдача решений преподавателю;
- тестирование решений задач с использованием системы NSUts;
- итоговый контроль в форме устного диффзачета.

Индивидуальные задания занесены в систему тестирования NSUts в виде туров. Для каждой задачи подготовлен набор тестов для проверки правильности решений. Туры проводятся по темам лекций, а также по решению задач на сторонних российских и международных серверах. Лабораторные работы всегда сопровождается итоговым анализом достигнутого, обсуждением эффективности выполненных задач, а также границ применимости обрабатываемых методов и приемов. Как следствие, у студентов появляется возможность критического осмысления знаний и навыков, полученных по темам курса.

**Промежуточная аттестация** (итоговая по дисциплине) осуществляется в форме диффзачета. Итоговая оценка выставляется по результатам выполненного портфолио.

В таблице 6.1 представлено соответствие форм аттестации заявляемым требованиям к результатам освоения дисциплины.

Таблица 6.1

Коды компетенций	Результаты обучения	1 семестр	2 семестр
		Дифф.зачет	Дифф.зачет
УК-2	УК-2.2 Уметь: проводить анализ	+	+

	поставленной цели и формулировать задачи, которые необходимо решить для ее достижения; анализировать альтернативные варианты решений для достижения намеченных результатов; использовать нормативно- правовую документацию в сфере профессиональной деятельности		
<b>ОПК-1</b>	<b>ОПК-1.2</b> Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и обще-инженерных знаний, методов математического анализа и моделирования	+	+
<b>ОПК-8</b>	<b>ОПК-8.1</b> Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения	+	+
<b>ОПК-8</b>	<b>ОПК-8.2</b> Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули	+	+
<b>ОПК-8</b>	<b>ОПК-8.3</b> Владеть: языком программирования; навыками отладки и тестирования работоспособности программы	+	+
<b>ОПК-9</b>	<b>ОПК-9.2</b> Уметь: находить и анализировать техническую документацию по использованию программного средства, выбирать и использовать необходимые функции программных средств для решения конкретной задачи	+	+

Требования к структуре и содержанию портфолио, оценочные средства, а также критерии оценки сформированности компетенций и освоения дисциплины в целом, представлены в Фонде оценочных средств, являющемся приложением 1 к настоящей рабочей программе дисциплины.

## 7. Литература

1. Вирт, Никлаус. Алгоритмы и структуры данных / Н. Вирт ; пер. с англ. Д.Б. Подшивалова. Москва : Мир, 1989. 360 с. : ил. ; 20 см. ISBN 5-03-001045-9. (24 экз)

№ п/п	Наименование Интернет-ресурса	Краткое описание
1	Система тестирования NSUts [Электронный ресурс] URL: <a href="https://olympic.nsu.ru/nsuts-new/login.cgi">https://olympic.nsu.ru/nsuts-new/login.cgi</a>	Автоматическая система тестирования, в которой собраны индивидуальные задания и тесты для лабораторных работ. По некоторым разделам приведена теория.

## 8. Учебно-методическое и программное обеспечение дисциплины

### 8.1. Учебно-методическое обеспечение

1. Чурина Т.Г., Нестеренко Т.В. Методы программирования: алгоритмы и структуры данных. Часть 3. Динамические структуры данных, алгоритмы на графах: учеб. Пособие/Новосиб. гос. ун-т.– Новосибирск: РИЦ НГУ, 2014. ISBN 978-5-4437-0278-0, 216 с. (60 экз)

### 8.2. Программное обеспечение

Для обеспечения реализации дисциплины используется стандартный комплект программного обеспечения (ПО), включающий регулярно обновляемое лицензионное ПО Windows и MS Office.

Перечень специализированного программного обеспечения для изучения дисциплины представлен в таблице 8.1.

#### Специализированное программное обеспечение

Таблица 8.1

№	Наименование ПО	Назначение
1	Microsoft Visual Studio 2019	Среда разработки приложений

## 9. Профессиональные базы данных и информационные справочные системы

1. Полнотекстовые журналы Springer Journals за 1997-2015 г., электронные книги (2005-2016 гг.), коллекция научных биомедицинских и биологических протоколов SpringerProtocols, коллекция научных материалов в области физических наук и инжиниринга SpringerMaterials, реферативная БД по чистой и прикладной математике zbMATH.

2. Электронная библиотека диссертаций Российской государственной библиотеки (ЭБД РГБ)

3. Лицензионные материалы на сайте eLibrary.ru

## 10. Материально-техническое обеспечение

Таблица 10.1

№	Наименование	Назначение
1	Презентационное оборудование (мультимедиа-проектор, экран, компьютер для управления)	Для проведения практических занятий
2	Компьютерный класс (с выходом в Internet)	Для проведения лабораторных занятий и организации самостоятельной работы

Материально-техническое обеспечение образовательного процесса по дисциплине для обучающихся из числа лиц с ограниченными возможностями здоровья осуществляется согласно «Порядку организации и осуществления образовательной деятельности по образовательным программам для инвалидов и лиц с ограниченными возможностями здоровья в Новосибирском государственном университете».



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования «Новосибирский национальный исследовательский  
государственный университет» (Новосибирский государственный университет,  
НГУ)

**Факультет информационных технологий**

СОГЛАСОВАНО

Декан ФИТ НГУ



М.М. Лаврентьев

«03» июля 2019 г.

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**  
**по дисциплине Алгоритмы и структуры данных**

Направление подготовки: 09.03.01 ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА

Направленность (профиль): Программная инженерия и компьютерные науки

Квалификация: бакалавр

Форма обучения: очная

Год обучения: 1, семестр 1, 2

Форма аттестации	Семестр
Дифференцированный зачет	1
Дифференцированный зачет	2

**Фонд оценочных средств** промежуточной аттестации по дисциплине является **Приложением 1** к рабочей программе дисциплины «Алгоритмы и структуры данных», реализуемой в рамках образовательной программы высшего образования – программы бакалавриата 09.03.01 Информатика и вычислительная техника, направленность (профиль): Программная инженерия и компьютерные науки.

Фонд оценочных средств промежуточной аттестации по дисциплине утвержден решением ученого совета факультета информационных технологий, протокол № 75 от 02.07.2019.

Разработчики:

*доцент* кафедры Общей информатики ФИТ,  
*кандидат физико-математических наук*

 Т.Г. Чурина

Заведующий кафедрой Систем информатики ФИТ,  
доктор физико-математических наук

 М.М. Лаврентьев

Ответственный за образовательную программу:

*доцент* кафедры систем информатики ФИТ,  
*кандидат технических наук*

 А.А. Романенко

## 1. Содержание и порядок проведения промежуточной аттестации по дисциплине

### 1.1. Общая характеристика содержания промежуточной аттестации

Промежуточная аттестация по дисциплине «Алгоритмы и структуры данных», проводится по завершению периода освоения образовательной программы (семестра) для оценки сформированности компетенций в части следующих индикаторов достижения компетенции (таблица П1.1).

Таблица П1.1

Код	Компетенции, формируемые в рамках дисциплины «Алгоритмы и структуры данных»	Семестр 1	Семестр 2
		диф.зачет	диф.зачет
<b>УК-2</b> Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений в части следующих индикаторов достижения компетенции:			
<b>УК-2.2</b>	Уметь: проводить анализ поставленной цели и формулировать задачи, которые необходимо решить для ее достижения; анализировать альтернативные варианты решений для достижения намеченных результатов; использовать нормативно- правовую документацию в сфере профессиональной деятельности	+	+
<b>ОПК-1</b> Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности			
<b>ОПК-1.2</b>	<b>ОПК-1.2</b> Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и обще-инженерных знаний, методов математического анализа и моделирования.	+	+
<b>ОПК-8</b> Способен разрабатывать алгоритмы и программы, пригодные для практического применения			
<b>ОПК-8.1</b>	<b>ОПК-8.1</b> Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения	+	+
<b>ОПК-8.2</b>	<b>ОПК-8.2</b> Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули	+	+
<b>ОПК-8.3</b>	<b>ОПК-8.3</b> Владеть: языком программирования; навыками отладки и тестирования работоспособности программы	+	+
<b>ОПК-9</b> Способен осваивать методики использования программных средств для решения задач			
<b>ОПК-9.2</b>	<b>ОПК-9.2</b> Уметь: находить и анализировать техническую документацию по использованию программного средства, выбирать и использовать необходимые функции программных средств для решения конкретной задачи	+	+

## 1.2. Порядок проведения промежуточной аттестации по дисциплине

Все компетенции, формируемые в рамках дисциплины, оцениваются через портфолио.

Портфолио включает следующие темы.

Алгоритмы перевода из систем счисления с различными основаниями, способы реализации длинной арифметики, понятие маски и решение задач с их использованием, проблемы кодирования, алгоритмы генерации перестановок, поиска, сортировки, хэш-функции. Алгоритм Евклида нахождение наибольшего общего делителя, диофантово уравнение, наименьшее общее кратное.

Основные определения и способы представления графов и деревьев, способы реализации классических задач на графах и деревьях: транзитивное замыкание, кратчайшие пути, достижимость вершин, топологическая сортировка, поиск двусвязных компонент в граф, алгоритмы построения минимальных остовных деревьев Прима, Краскала, задачи о лабиринтах, эйлеровы циклы.

NP-полные задачи, способы сведения задачи к известной. Способы реализации: полный перебор, перебор с отсечением, метод ветвей и границ. Метод динамического программирования. Классические задачи и задачи с международных чемпионатов, оценка быстродействия.

Стратегии игр. Приемы решения геометрических задач. Задачи теории расписаний. Потoki в сетях и паросочетания, поиск максимальных паросочетаний в двудольном графе.

## 2. Требования к структуре и содержанию фонда оценочных средств промежуточной аттестации по дисциплине

Перечень оценочных средств, применяемых на каждом этапе проведения промежуточной аттестации по дисциплине, представлен в таблице П1.3.

Таблица П1.3

№ п/п	Наименование оценочного средства	Краткая характеристика оценочного средства	Представление оценочного средства в фонде
<b>Семестр 1</b>			
<b>Этап 1 - дифзачет</b>			
1	Портфолио	Решение задач по темам дисциплины и сдача их в автоматическую систему тестирования NSUts и преподавателю. При сдаче заданий в систему NSUts Необходимо следовать методическим указаниям, представленными в системе NSUts.	Комплект разноуровневых заданий
<b>Семестр 2</b>			
<b>Этап 2 - дифзачет</b>			
2	Портфолио	Решение задач по темам дисциплины и сдача их в автоматическую систему тестирования NSUts и преподавателю	Комплект разноуровневых заданий

### 2.1 Требования к структуре и содержанию оценочных средств аттестации

#### 2.1.1 Описание оценочного средства

### Структура и содержание портфолио

Портфолио должно содержать не менее 12 заданий в течение семестра.

Каждое выполненное задание состоит из 5 решенных задач, сданных в систему тестирования NSUts.

Оценка за портфолио формируется из оценок выполнения лабораторных заданий, сданных в систему NSUts.

Для получения положительной оценки необходимо сдать все задания.

Оценка за портфолио выставляется согласно следующим критериям

<b>Оценка</b>	<b>Критерий</b>
«отлично»	среднее значение балла $\geq 4.5$
«хорошо»	среднее значение балла $\geq 3.5$ , но $< 4.5$
«удовлетворительно»	среднее значение балла $< 3.5$
«неудовлетворительно»	одно или несколько заданий не сданы

**3. Критерии оценки сформированности компетенций в рамках промежуточной аттестации по дисциплине**

Таблица П1.5

<b>Результат обучения</b>	<b>Структурные элементы оценочных средств</b>	<b>Показатель сформированности</b>	<b>Не сформирован (неудовл., 2 балла)</b>	<b>Пороговый уровень (удовл., 3 балла)</b>	<b>Базовый уровень (хорошо, 4 балла)</b>	<b>Продвинутый уровень (отлично, 5 баллов)</b>
<b>УК-2</b>	Портфолио	<b>УК-2.2</b> Уметь: проводить анализ поставленной цели и формулировать задачи, которые необходимо решить для ее достижения; анализировать альтернативные варианты решений для достижения намеченных результатов; использовать нормативно-правовую документацию в сфере профессиональной деятельности	Знания отсутствуют или носят поверхностный характер; студент слабо ориентируется в базовых понятиях, допускает грубые ошибки.	Знания присутствуют, но содержат отдельные пробелы; студент в целом ориентируется в базовых объектах дисциплины.	Знания присутствуют, студент в целом владеет материалом курса, ответ содержит отдельные недочеты.	Знания присутствуют, студент в полной мере владеет материалом курса.
<b>ОПК-1</b>	Портфолио	<b>ОПК1.2</b> Уметь: решать стандартные профессиональные задачи с применением естественнонаучных и обще-инженерных знаний, методов математического анализа и моделирования	Умение отсутствует или носит фрагментарный характер; студент допускает грубые фактические и/или методологические ошибки.	Умение присутствует, но студент испытывает затруднения при его применении, допускает ошибки, нуждается в подсказках.	Умение в целом сформировано; студент в состоянии его применить к указанным вопросам диффзачета и продемонстрировать в системе NSUts, ответ содержит отдельные недочеты.	Умение сформировано, студент владеет языком программирования и навыками отладки и тестирования работоспособности программы

<b>ОПК-8</b>	Портфолио	<b>ОПК-8.1</b> Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения	Знания отсутствуют или носят поверхностный характер; студент слабо ориентируется в базовых понятиях, допускает грубые ошибки.	Знания присутствуют, но содержат отдельные пробелы; студент в целом ориентируется в базовых объектах дисциплины.	Знания присутствуют, студент в целом владеет материалом курса, ответ содержит отдельные недочеты.	Знания присутствуют, студент в полной мере владеет материалом курса.
<b>ОПК-8</b>	Портфолио	<b>ОПК-8.2</b> – Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули	Умение отсутствует или носит фрагментарный характер; студент допускает грубые ошибки.	Умение присутствует, но содержит пробелы; студент испытывает затруднения при его применении, допускает ошибки, нуждается в подсказках.	Умение в целом сформировано; студент в состоянии его применять и продемонстрировать в системе NSUts, ответ содержит отдельные недочеты.	Умение сформировано, студент в состоянии успешно его применять.
<b>ОПК-8</b>	Портфолио	<b>ОПК-8.3</b> – Владеть: языком программирования; навыками отладки и тестирования работоспособности программы	Умение отсутствует или носит фрагментарный характер; студент допускает грубые фактические и/или методологические ошибки.	Умение присутствует, но студент испытывает затруднения при его применении, допускает ошибки, нуждается в подсказках.	Умение в целом сформировано; студент в состоянии его применять к указанным вопросам диффзачета и продемонстрировать в системе NSUts, ответ содержит отдельные недочеты.	Умение сформировано, студент владеет языком программирования и навыками отладки и тестирования работоспособности программы
<b>ОПК-9</b>	Портфолио	<b>ОПК-9.2</b> Уметь: находить и анализировать техническую	Умение работать в системе тестирования NSUts от-	Умение работать в системе тестирования	Умение работать в системе тестирования NSUts в целом	Умение работать в системе тестирования NSUts сформиро-

		документацию по использованию программного средства, выбирать и использовать необходимые функции программных средств для решения конкретной задачи	сутствует или носит фрагментарный характер; студент допускает грубые фактические и/или методологические ошибки.	NSUts присутствует, но студент испытывает затруднения при его применении, допускает ошибки, нуждается в подсказках.	сформировано; студент в состоянии его применять к указанным вопросам диффзачета и продемонстрировать в системе NSUts, ответ содержит отдельные недочеты.	вано, студент владеет языком программирования и навыками отладки и тестирования работоспособности программы
--	--	--	---	---	--	---

#### 4. Критерии выставления оценок по результатам промежуточной аттестации по дисциплине

Результаты промежуточной аттестации в первом семестре определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно» Оценки «отлично», «хорошо», «удовлетворительно» (или «зачтено») означают успешное прохождение промежуточной аттестации.

Результаты промежуточной аттестации во втором семестре определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно» Оценки «отлично», «хорошо», «удовлетворительно» (или «зачтено») означают успешное прохождение промежуточной аттестации.

Диф.зачет проводится по результатам оценивания портфолио студента.

<b>Оценка</b>	<b>Критерий</b>
«отлично»	все проверяемые результаты обучения сформированы на уровне 4 или 5 баллов; среднее значение балла $\geq 4.5$
«хорошо»	все проверяемые результаты обучения сформированы на уровне не ниже порогового (3 балла); среднее значение балла $\geq 3.5$ , но $< 4.5$
«удовлетворительно»	все проверяемые результаты обучения сформированы на уровне не ниже порогового (3 балла); среднее значение балла $< 3.5$
«неудовлетворительно»	один или несколько проверяемых результатов обучения не сформированы

