

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
"Казанский (Приволжский) федеральный университет"  
Институт вычислительной математики и информационных технологий



УТВЕРЖДАЮ  
Проректор по образовательной деятельности КФУ  
Проф. Д.А. Таюрский

\_\_\_\_\_» \_\_\_\_\_ 20\_\_ г.

подписано электронно-цифровой подписью

**Программа дисциплины**  
**Алгоритмы и структуры данных Б1.В.ОД.13**

Направление подготовки: 38.03.05 - Бизнес-информатика

Профиль подготовки: не предусмотрено

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

**Автор(ы):**

Фофанов В.Б.

**Рецензент(ы):**

Тагиров Р.Р. , Миссаров М.Д.

**СОГЛАСОВАНО:**

Заведующий(ая) кафедрой: Миссаров М. Д.

Протокол заседания кафедры No \_\_\_\_ от " \_\_\_\_ " \_\_\_\_\_ 201\_\_ г

Учебно-методическая комиссия Института вычислительной математики и информационных технологий:

Протокол заседания УМК No \_\_\_\_ от " \_\_\_\_ " \_\_\_\_\_ 201\_\_ г

Регистрационный No 9147819

## Содержание

1. Цели освоения дисциплины
2. Место дисциплины в структуре основной образовательной программы
3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля
4. Структура и содержание дисциплины/ модуля
5. Образовательные технологии, включая интерактивные формы обучения
6. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины и учебно-методическое обеспечение самостоятельной работы студентов
7. Литература
8. Интернет-ресурсы
9. Материально-техническое обеспечение дисциплины/модуля согласно утвержденному учебному плану

Программу дисциплины разработал(а)(и) Фофанов В.Б.

### 1. Цели освоения дисциплины

Данная учебная дисциплина реализуется как цикл лекционных и лабораторных занятий, которые знакомят студентов с основами применения при решении различных задач структур данных различной сложности (массивы, списки, хэш-таблицы, деревья, графы, стеки, очереди) и алгоритмов работы с ними. Для решения различных практических задач используется язык программирования C#.

Данный курс должен сформировать у студентов навыки обоснованного выбора способа хранения данных при решении задач обработки больших объемов информации, что может сделать это решение эффективным и конкурентоспособным.

### 2. Место дисциплины в структуре основной образовательной программы высшего профессионального образования

Данная учебная дисциплина включена в раздел "Б1.В.ОД.13 Дисциплины (модули)" основной образовательной программы 38.03.05 Бизнес-информатика и относится к обязательным дисциплинам. Осваивается на 1 курсе, 2 семестр.

Дисциплина "Алгоритмы и структуры данных" изучается на 1 курсе во 2 семестре обучения бакалавров. Предварительно студенты должны изучить дисциплины "Теоретические основы информатики" и "Программирование". Знания по этому курсу требуются в дальнейшем при изучении учебных, "Объектно-ориентированный анализ и программирование", "Вычислительные системы и телекоммуникации", "Проектирование информационных систем" и других учебных дисциплин, которые предусмотрены учебным планом по направлению "Бизнес-информатика". Навыки, полученные при изучении этого предмета, будут использованы студентами при написании курсовых и дипломных работ.

### 3. Компетенции обучающегося, формируемые в результате освоения дисциплины /модуля

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОК-12	осознает сущность и значение информации в развитии современного общества; владеет основными методами, способами и средствами получения, хранения, переработки информации
ОК-16	способен работать с информацией из различных источников
ПК-20 (профессиональные компетенции)	использовать соответствующий математический аппарат и инструментальные средства для обработки, анализа и систематизации информации по теме исследования
ПК-21 (профессиональные компетенции)	готовить научно-технические отчеты, презентации, научные публикации по результатам выполненных исследований

В результате освоения дисциплины студент:

1. должен знать:

- основные виды структур данных, применяемых при решении задач;
- алгоритмы обработки информации, хранящейся в различных видах структур данных;
- достоинства и недостатки каждого вида структур данных для применения при решении различных задач.

2. должен уметь:

- делать обоснованный выбор используемых при решении задач структур данных;
- применять структуры данных и алгоритмы их обработки при решении различных задач.

3. должен владеть:

- навыками создания собственных и использования библиотечных структур данных при разработке программ на языке программирования C#.

4. должен демонстрировать способность и готовность:

- применять полученные знания и навыки в своей дальнейшей профессиональной деятельности.

#### 4. Структура и содержание дисциплины/ модуля

Общая трудоемкость дисциплины составляет 6 зачетных(ые) единиц(ы) 216 часа(ов).

Форма промежуточного контроля дисциплины: экзамен во 2 семестре.

Суммарно по дисциплине можно получить 100 баллов, из них текущая работа оценивается в 50 баллов, итоговая форма контроля - в 50 баллов. Минимальное количество для допуска к зачету 28 баллов.

86 баллов и более - "отлично" (отл.);

71-85 баллов - "хорошо" (хор.);

55-70 баллов - "удовлетворительно" (удов.);

54 балла и менее - "неудовлетворительно" (неуд.).

#### 4.1 Структура и содержание аудиторной работы по дисциплине/ модулю

##### Тематический план дисциплины/модуля

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи- ческие занятия	Лабора- торные работы	
1.	Тема 1. Базовые алгоритмы решения задач.	2	1-3	6	0	6	Письменное домашнее задание
2.	Тема 2. Динамические структуры данных.	2	4-7	8	0	6	Письменное домашнее задание
3.	Тема 3. Граф как структура данных.	2	8-10	6	0	6	Письменное домашнее задание
4.	Тема 4. Деревья как частный случай графов.	2	11-12	4	0	6	Письменное домашнее задание

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практи- ческие занятия	Лабора- торные работы	
5.	Тема 5. Деревья сортировки и сбалансированные деревья.	2	13-16	8	0	6	Письменное домашнее задание
6.	Тема 6. В-деревья.	2	17-18	4	0	6	Письменное домашнее задание
.	Тема . Итоговая форма контроля	2		0	0	0	Экзамен
	Итого			36	0	36	

## 4.2 Содержание дисциплины

### Тема 1. Базовые алгоритмы решения задач.

#### *лекционное занятие (6 часа(ов)):*

Алгоритмы поиска. Алгоритмы сортировки: внутренняя и внешняя сортировка. Использование хеширования для поиска данных. Решение задач с перебором: алгоритмы типа "разделяй и властвуй", метод динамического программирования, метод ветвей и границ.

#### *лабораторная работа (6 часа(ов)):*

Решение задач на тему "Сортировка и поиск" и "Метод ветвей и границ и динамическое программирование"

### Тема 2. Динамические структуры данных.

#### *лекционное занятие (8 часа(ов)):*

Определение структуры данных. Линейные связанные списки: однонаправленные и двунаправленные. Очередь, стек, дек - реализации в виде массива и списка. Примеры приложений, использующих списки, стеки и очереди.

#### *лабораторная работа (6 часа(ов)):*

Решение задач на темы использования динамических структур данных, особенно стеков и очередей.

### Тема 3. Граф как структура данных.

#### *лекционное занятие (6 часа(ов)):*

Граф как структура данных. Основные определения теории графов. Приложения, использующие графы как структуры данных. Представления графов в программах. Алгоритмы обхода графов : поиск в глубину и поиск в ширину. Алгоритмы поиска кратчайших путей : алгоритм Флойда и алгоритм Дейкстры. Построение кратчайших остовов : алгоритм Краскала.

#### *лабораторная работа (6 часа(ов)):*

Решение задач на тему "Графы". Хранение графа в программе, проверки различных условий (полнота, связность, двудольность), поиск кратчайших путей и остовов минимального веса.

### Тема 4. Деревья как частный случай графов.

#### *лекционное занятие (4 часа(ов)):*

Определения ориентированного, упорядоченного, бинарного дерева. Представление деревьев в программе. Код Прюфера. Представление упорядоченных ориентированных деревьев. Представление бинарных деревьев.

#### *лабораторная работа (6 часа(ов)):*

Решение задач на тему "Деревья". Использование деревьев для поиска.

**Тема 5. Деревья сортировки и сбалансированные деревья.****лекционное занятие (8 часа(ов)):**

Определение дерева сортировки, приложения использования. Алгоритм поиска в дереве сортировки. Алгоритм вставки в дерево сортировки. Алгоритм удаления из дерева сортировки. Определение сбалансированного дерева. Балансировка деревьев.

**лабораторная работа (6 часа(ов)):**

Решение задач на тему "Деревья сортировки". Операции с деревьями сортировки, использование. Операции по балансировке деревьев.

**Тема 6. В-деревья.****лекционное занятие (4 часа(ов)):**

Определение В-дерева. Алгоритмы поиска в В-дереве. Алгоритм вставки в В-дерево. Алгоритм удаления из В-дерева.

**лабораторная работа (6 часа(ов)):**

Решение задач по теме "В-деревья". Реализация основных операций с В-деревьями и использованию деревьев.

**4.3 Структура и содержание самостоятельной работы дисциплины (модуля)**

№	Раздел дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
1.	Тема 1. Базовые алгоритмы решения задач.	2	1-3	подготовка домашнего задания	14	домашнее задание
2.	Тема 2. Динамические структуры данных.	2	4-7	подготовка домашнего задания	16	домашнее задание
3.	Тема 3. Граф как структура данных.	2	8-10	подготовка домашнего задания	16	домашнее задание
4.	Тема 4. Деревья как частный случай графов.	2	11-12	подготовка домашнего задания	14	домашнее задание
5.	Тема 5. Деревья сортировки и сбалансированные деревья.	2	13-16	подготовка домашнего задания	16	домашнее задание
6.	Тема 6. В-деревья.	2	17-18	подготовка домашнего задания	14	домашнее задание
	Итого				90	

**5. Образовательные технологии, включая интерактивные формы обучения**

Дисциплина представляет собой цикл лекционных и лабораторных (практических) занятий. Лабораторные занятия посвящены выработке навыков разработки программ на языке программирования C# с использованием собственных и стандартных структур данных различных видов. Лабораторные занятия проходят в компьютерных классах с использованием интерактивной доски для наглядного представления алгоритмов и разработки программ на всех этапах ее создания и компиляции. Лабораторные занятия проходят в интерактивной форме обсуждения решения различных задач или в активной форме самостоятельного решения задач студентами. Контроль за выполнением самостоятельной работы проявляется в функциональном тестировании выполненных студентами заданий на примерах, предложенных преподавателем.

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

### **Тема 1. Базовые алгоритмы решения задач.**

домашнее задание , примерные вопросы:

Углубленное изучение литературы. Обсуждение. Решение задач на тему "Сортировка и поиск", "Рекурсивные алгоритмы поиска - динамическое программирование и метод ветвей и границ"

### **Тема 2. Динамические структуры данных.**

домашнее задание , примерные вопросы:

Углубленное изучение литературы. Обсуждение. Решение задач на тему "Динамические структуры данных - списки, стеки, очереди".

### **Тема 3. Граф как структура данных.**

домашнее задание , примерные вопросы:

Углубленное изучение литературы. Обсуждение. Решение задач на тему "Граф. Поиск кратчайших путей и кратчайших остовов"

### **Тема 4. Деревья как частный случай графов.**

домашнее задание , примерные вопросы:

Углубленное изучение литературы. Обсуждение. Решение задач на тему "Деревья".

### **Тема 5. Деревья сортировки и сбалансированные деревья.**

домашнее задание , примерные вопросы:

Углубленное изучение литературы. Обсуждение. Решение задач на тему "Деревья сортировки"

### **Тема 6. В-деревья.**

домашнее задание , примерные вопросы:

Углубленное изучение литературы. Обсуждение. Решение задач на тему "В-деревья"

### **Итоговая форма контроля**

экзамен (в 2 семестре)

Примерные вопросы к итоговой форме контроля

По данной дисциплине предусмотрен экзамен, который проводится в форме решения задач на компьютере. Примерные варианты экзаменационных задач приведены в Приложении 1:

1. Даны множества  $A$  и  $B$  из  $n_A$  и  $n_B$  целых положительных чисел (каждое число входит в соответствующее множество только один раз). Создать функцию `TestInclu()` для проверки принадлежности множества  $A$  множеству  $B$ . Функция возвращает 1, если  $A$  содержится в  $B$ , и 0 в противном случае.

Пример:  $A=\{3,4,1,6,7\}$ ,  $n_A=5$ ,  $B=\{8,1,3,10,\}$ ,  $n_B=4$ . Результат: `TestInclu()` возвращает 0.

2. Даны множества  $A$  и  $B$  из  $n_A$  и  $n_B$  целых положительных чисел (каждое число входит в соответствующее множество только один раз). Создать функцию `Union()`, которая строит их объединение  $A+B$  и возвращает адрес массива.

3. Даны множества  $A$  и  $B$  из  $n_A$  и  $n_B$  целых положительных чисел (каждое число входит в соответствующее множество только один раз). Создать функцию с именем `Intersection` для получения их пересечения  $A*B$  (в  $A*B$  не должно быть повторяющихся чисел).  
Пример.  $A=\{3,4,1,6,7\}$ ,  $n_A=5$ ,  $B=\{8,1,3,10\}$ ,  $n_B=4$ . Пересечение  $A*B=\{3,1\}$ ,  $n_{A*B}=2$ .
4. Даны множества  $A$  и  $B$  из  $n_A$  и  $n_B$  целых положительных чисел (каждое число входит в соответствующее множество только один раз). Создать функцию `Difference()` для получения их разности  $A\setminus B$  (в  $A\setminus B$  не должно быть повторяющихся чисел).  
Пример:  $A=\{3,4,1,6,7\}$ ,  $n_A=5$ ,  $B=\{8,1,3,10\}$ ,  $n_B=4$ . Разность  $A\setminus B=\{4,6,7\}$ .
5. Дана последовательность  $A=(a_j)_{1 \leq j \leq n}$  из  $n$  целых чисел. Создать функцию с именем `Count`, возвращающую количество чисел, которые встречаются в  $A$  только один раз.  
Пример:  $A=\{0,3,8,3,1,9,0,1,1\}$ . Ответ: 2.
6. Дана последовательность  $A=(a_j)_{1 \leq j \leq n}$  из  $n$  целых чисел. Создать функцию с именем `Count`, возвращающую количество чисел, которые встречаются в  $A$  два или большее число раз. В главной функции создать массив с подпоследовательностью  $A$ , подсчитать количество повторяющихся чисел и отобразить его на дисплее.  
Пример:  $A=\{0,3,8,3,1,9,0,1,1\}$ . Ответ: 3.
7. Для хранения многочлена  $P(x)=a_0+a_1x+\dots+a_nx^n$  степени  $n$  используется массив. Определить функцию `AddPolynom()` для сложения двух многочленов.
8. Пусть  $A=(a_j)_{1 \leq j \leq n}$  - последовательность из  $n$  целых чисел. Число  $a_k$ ,  $1 < k < n$ , назовем локальным максимумом, если  $a_{k-1} < a_k > a_{k+1}$ . Определить функцию `IndexMax()`, вычисляющую номер наибольшего локального максимума в  $A$ . При отсутствии локальных максимумов функция должна вернуть -1.  
Пример.  $A=\{0,4,3,2,2,8,7\}$ . Ответ: 5 (нумерация начинается с нуля).
9. (Слияние последовательностей) Пусть  $A=(a_i)_{1 \leq i \leq n_A}$  и  $B=(b_i)_{1 \leq i \leq n_B}$  - неубывающие последовательности из  $n_A$  и  $n_B$  целых чисел соответственно. Создать функцию `Merger()`, осуществляющую построение неубывающей последовательности  $C$  длины  $n_A+n_B$  из членов  $A$  и  $B$  и возвращающую адрес массива, в котором хранится  $C$ .  
В главной функции создать массивы с  $A$  и  $B$ , получить с помощью `Merger()` массив  $C$  и отобразить его на дисплее.  
Пример.  $A=\{1,4,5,6\}$ ,  $n_A=4$ ,  $B=\{3,5,7\}$ ,  $n_B=3$ .  $C=\{1,3,4,5,5,6,7\}$ ,  $n_C=7$ .
10. Пусть  $A=(a_j)_{1 \leq j \leq n}$  - последовательность из  $n$  положительных целых чисел,  $x$  - целое число. Создать функцию `Test()` для проверки существования подпоследовательности  $a_{start}, a_{start+1}, \dots, a_{finish}$ ,  $1 \leq start \leq finish \leq n$ , такой, что  $a_{start} + a_{start+1} + \dots + a_{finish} = x$ . Если подпоследовательность существует, то функция возвращает 1,  $start$  и  $finish$ , в противном случае - 0.  
В главной функции определить  $A$ ,  $x$  и проверить с помощью `Test()` наличие соответствующей подпоследовательности в  $A$ .
11. (Неубывающие цепочки) Пусть  $A=(a_j)_{1 \leq j \leq n}$  - последовательность из  $n$  положительных целых чисел. Ее подпоследовательность  $a_{m+1}, a_{m+2}, \dots, a_{m+k}$  называется неубывающей цепочкой длины  $k$ , если  $a_{m+1} \leq a_{m+2} \leq \dots \leq a_{m+k}$  и  $a_m > a_{m+1}$ ,  $a_{m+k} > a_{m+k+1}$ . При этом  $a_1$  всегда является началом цепочки, а  $a_n$  - окончанием или началом. Определить функцию `CountSubSet()`, получающую вектор  $B=(b_j)_{1 \leq j \leq n}$ , у которого  $b_j$  является числом неубывающих цепочек длины  $j$ .  
Пример:  $A=\{10, 3, 3, 15, 14, 21\}$ ,  $n=6$ .  $B=\{1, 1, 1, 0, 0, 0\}$ .
12. Задано вещественно (типа `double`) число  $x$  и полное имя двоичного файла, в котором хранится таблица вещественных чисел из  $h$  строк и  $w$  столбцов, записанная с помощью стандартной функции `fwrite()`. Определить функцию с именем `Replaceltem()`, выполняющую замену в файле элемента из последней строки и последнего столбца числом  $x$ .



13. Дано полное имя файла, в котором хранятся векторы разной длины с вещественными (типа double) координатами, записанные с помощью стандартной функции fprintf(). При записи вектора вначале записывается его длина (тип int), а затем его вещественные координаты. Определить функцию SumItem(), которая вычисляет сумму координат вектора с указанным порядковым номером.

14. Дано полное имя файла, в котором хранятся векторы разной длины с вещественными (типа double) координатами, записанные с помощью стандартной функции fprintf(). При записи вектора вначале записывается его длина (тип int), а затем его вещественные координаты. Определить функцию AddVector(), которая записывает в конец данного файла вектор, длина и координаты которого вводятся с клавиатуры.

15. Дано полное имя файла, в котором хранится таблица вещественных (типа double) чисел, записанная с помощью стандартной функции fprintf(). Вначале записана ее высота h и ширина w (тип int), а затем - элементы. Определить функцию GetColumn(), которая читает из файла столбец с указанным номером j и сохраняет его в массиве.

16. Дано полное имя файла, в котором хранится таблица вещественных (типа double) чисел, записанная с помощью стандартной функции fwrite(). Вначале записана ее высота h и ширина w (тип int), а затем - элементы. Создать функцию ReplaseLine(), осуществляющей замену элементов строки с минимальной суммой элементов нулями.

17. Задан полный путь к файлу с таблицей вещественных (типа double) чисел, записанной с помощью стандартной функции fwrite(). Вначале записана ее высота h и ширина w (тип int), а затем - элементы. Определить функцию Invert(), которая получает файл с таблицей, транспонирует таблицу и записывает ее в другой файл.

18. Текст - последовательность символов, состоящая из букв и пробелов. Слово - подпоследовательность, не содержащая пробелов. Тексты и слова хранятся в символьных строках. Создать функцию IndexWord(), обеспечивающую проверку присутствия в тексте заданного слова и вычисляющую позицию (от начала текста) первой буквы первого вхождения этого слова. При отсутствии слова в тексте функция возвращает -1.

19. Написать собственный вариант MyStrCmp() стандартной функции strcmp().

20. Текст - последовательность из букв и пробелов, хранящаяся в символьной строке. Создать функцию InsertWord(), осуществляющую вставку подпоследовательности символов (букв и пробелов) в текст, начиная с указанной позиции. Функция должна возвращать адрес символьной строки с измененным текстом.

Пример. Текст: "abcd efg h ijklmno", слово: "word", позиция: 2 (нумерация начинается с нуля).  
Результат: "abwordcd efg h ijklmno".

21. Текст - последовательность из букв и пробелов, хранящаяся в символьной строке. Создать функцию ExluWord(), осуществляющую исключение из текста подпоследовательности символов (букв и пробелов), начинающейся с позиции start и заканчивающейся в позиции finish (start<=finish). Функция должна возвращать адрес символьной строки с измененным текстом.

В главной функции создать символьную строку с текстом, выбрать в тексте подпоследовательность символов, задав ее начало и конец, исключить ее из текста, используя функцию ExcluWord(), и отобразить полученный текст на дисплее..

Пример. Текст: "Сегодня хорошая погода," Start=0, finish=7. Текст после исключения: "хорошая погода."

Текущий контроль осуществляется посредством устного опроса и контрольных работ. Примерные варианты вопросов для текущего контроля приведены в Приложении2:

1. Определение класса
2. Виды членов класса
3. Области видимости членов класса
4. Операция доступа к членам класса
5. Статические члены класса
6. Встроенные классы int, double, char, byte
7. Класс Console

8. Класс Math
9. Возможности класса string
10. Массивы и возможности класса array
11. Виды конструкторов класса
12. Правила определения свойств класса
13. Использование индексов
14. Интерфейсы
15. Правила перегрузки операций
16. Наследование
17. Базовые и производные классы
18. Конструкторы производных классов
19. Определение делегатов
20. Вызов методов с помощью делегатов
21. Запись в файл объектов типа byte
22. Запись в файл чисел
22. Запись в файл текста
23. Сериализация и десериализация

### 7.1. Основная литература:

- 1) Долгов, А. И. Алгоритмизация прикладных задач [Электронный ресурс] : Уч. пособ / А. И. Долгов. - М. : Флинта, 2011. - 136 с. - ISBN 978-5-9765-0086-0. ? Режим доступа: <http://www.znaniyum.com/bookread.php?book=406093>
- 2) Структуры и алгоритмы обработки данных: Учебное пособие / В.Д. Колдаев. - М.: ИЦ РИОР: НИЦ ИНФРА-М, 2014. - 296 с.: 60x90 1/16. - (Высшее образование: Бакалавриат). (переплет) ISBN 978-5-369-01264-2-Режим доступа: <http://znaniyum.com/bookread.php?book=418290>

### 7.2. Дополнительная литература:

- 1) Алгоритмы и структуры данных. Новая версия для Оберона / Никлаус Вирт ; пер. с англ. под ред. д.ф.-м.н. Ткачева Ф. В. - Москва : ДМК Пресс, 2014. - 272 . : ил. ; 21 .- (Классика программирования) .- 1-е изд. 2001 .- Библиогр. в конце гл. - Предм. указ. : с. 270-272 .- ISBN 978-5-97060-011-5 ((в обл.)) , 200.
- 2) Гаврилова, И. В. Разработка приложений [Электронный ресурс] : учеб. пособие / И. В. Гаврилова. - 2-е изд., стер. - М.: ФЛИНТА, 2012. - 242 с. - ISBN 978-5-9765-1482-9? Режим доступа: <http://znaniyum.com/bookread.php?book=455037>
- 3) Алгоритмы и структуры данных: Учебник / Белов В.В., Чистякова В.И. - М.:КУРС, НИЦ ИНФРА-М, 2016. - 240 с.: 60x90 1/16. - (Бакалавриат) (Переплёт 7БЦ) ISBN 978-5-906818-25-6 - Режим доступа: <http://znaniyum.com/catalog/product/551224>
- 4) Алгоритмы и структуры данных (CDIO): Учебник / Царев Р.Ю., Прокопенко А.В. - Краснояр.:СФУ, 2016. - 204 с.: ISBN 978-5-7638-3388-1 - Режим доступа: <http://znaniyum.com/catalog/product/967108>
- 5) Структуры и алгоритмы компьютерной обработки данных. Лабораторный практикум: Учебное пособие / Засорин С.В., Ломтева О.А. - М.:КУРС, 2018. - 384 с.: 60x90 1/16. - (Бакалавриат) (Переплёт 7БЦ) ISBN 978-5-907064-14-0 - Режим доступа: <http://znaniyum.com/catalog/product/977719>

### 7.3. Интернет-ресурсы:

Википедия - <http://ru.wikipedia.org>

Интернет-портал образовательных ресурсов по ИТ - <http://www.intuit.ru>

Интернет-портал по языку программирования С# - <http://msdn.microsoft.com/ru-ru/vcsharp/>

Интернет-портал со статьями по алгоритмике и программированию - <http://algotlist.manual.ru/>

Электронная библиотека по техническим наукам - <http://techlibrary.ru>

## **8. Материально-техническое обеспечение дисциплины(модуля)**

Освоение дисциплины "Алгоритмы и структуры данных" предполагает использование следующего материально-технического обеспечения:

Компьютерные классы (9 классов) лаборатории малой вычислительной техники Института ВМ и ИТ, оснащенные мультимедийным оборудованием.

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 38.03.05 "Бизнес-информатика" и профилю подготовки не предусмотрено .

Автор(ы):

Фофанов В.Б. \_\_\_\_\_

"\_\_" \_\_\_\_\_ 201\_\_ г.

Рецензент(ы):

Тагиров Р.Р. \_\_\_\_\_

Миссаров М.Д. \_\_\_\_\_

"\_\_" \_\_\_\_\_ 201\_\_ г.