

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

"Казанский (Приволжский) федеральный университет"  
Институт математики и механики им. Н.И. Лобачевского



**УТВЕРЖДАЮ**

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

" " 20\_\_ г.

**Программа дисциплины**

Практикум по решению задач на компьютере Б1.В.Од.24

Направление подготовки: 44.03.05 - Педагогическое образование (с двумя профилями подготовки)

Профиль подготовки: Математика, информатика и информационные технологии

Квалификация выпускника: бакалавр

Форма обучения: очное

Язык обучения: русский

**Автор(ы):**

Галимьянов А.Ф. , Першагин М.Ю.

**Рецензент(ы):**

Авхадиев Ф.Г.

**СОГЛАСОВАНО:**

Заведующий(ая) кафедрой: Галимьянов А. Ф.

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

Учебно-методическая комиссия Института математики и механики им. Н.И. Лобачевского :

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

Регистрационный №

Казань  
2019

## 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) доцент, к.н. (доцент) Галимьянов А.Ф. Кафедра информационных систем отделение фундаментальной информатики и информационных технологий , Anis.Galimjanoff@kpfu.ru ; старший преподаватель, б/с Першагин М.Ю. Кафедра теории функций и приближений отделение математики , Michael.Pershagin@kpfu.ru

## 1. Цели освоения дисциплины

Ориентироваться в выборе языка программирования для решения конкретной задачи и уметь эффективно его применить

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

Данная учебная дисциплина включена в раздел "Б1.В.ОД.24 Дисциплины (модули)" основной образовательной программы 44.03.05 Педагогическое образование (с двумя профилями подготовки) и относится к обязательным дисциплинам. Осваивается на 4 курсе, 7 семестр.

Дисциплина изучается на 4 курсе после основных дисциплин по программированию и должен закрепить знания и навыки программирования на изученных языках

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

В результате освоения дисциплины формируются следующие компетенции:

Шифр компетенции	Расшифровка приобретаемой компетенции
СПК-13 (профессиональные компетенции)	способность создавать и использовать современные информационные и коммуникационные технологии для создания, формирования и администрирования электронных образовательных ресурсов
СПК-14 (профессиональные компетенции)	способность реализовывать аналитические и технологические решения в области программного обеспечения (системного, прикладного и инструментального) и компьютерной обработки информации
СПК-15 (профессиональные компетенции)	способность создавать и размещать информацию в компьютерной сети
СПК-16 (профессиональные компетенции)	способность ориентироваться в информационном потоке, использовать рациональные способы получения, преобразования, систематизации и хранения информации, актуализировать ее в необходимых ситуациях интеллектуально-познавательной деятельности, способен структурировать информацию, организовывать ее поиск и защиту

В результате освоения дисциплины студент:

### 1. должен знать:

Знать основные идеи, лежащие в основе компьютерных наук, их практическое применение и возможности; основные методы алгоритмизации прикладных задач математики, механики, физики и других наук; основные языки программирования C, Pascal, Phyton.

### 2. должен уметь:

находить, анализировать и контекстно-обрабатывать научно-техническую информацию с помощью компьютера; активно использовать компьютер в профессиональной и социально-бытовой сфере; создавать базы данных и использовать ресурсы Интернет.

3. должен владеть:

Владеть навыками применения методов компьютерных наук для различного класса задач, умением довести их до числа; базовыми знаниями в областях информатики и информационных технологий, навыками использования программных средств и работы в компьютерных сетях.

4. должен демонстрировать способность и готовность:

алгоритмизировать, запрограммировать и реализовать решение конкретной научной или прикладной задачи на компьютере

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

Общая трудоемкость дисциплины составляет 2 зачетных(ые) единиц(ы) 72 часа(ов).

Форма промежуточного контроля дисциплины: зачет в 7 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	Практические занятия	Лабораторные работы	
1.	Тема 1. Структуры данных на языках программирования. Внутренняя и внешняя сортировки	7	1-2	0	0	4	Лабораторные работы
2.	Тема 2. Сортировка числового массива методом пузырька. Сортировка массива объектов методом пузырька. Сортировка числового массива методом просеивания. Сортировка числового массива методом Шелла. Сортировка числового массива методом быстрой сортировки.	7	3-5	0	0	6	Лабораторные работы

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
3.	Тема 3. Метод естественного слияния.	7	6	0	0	2	Лабораторные работы
4.	Тема 4. Алгоритмы внешней сортировки.	7	7	0	0	2	Лабораторные работы
5.	Тема 5. Алгоритмы поиска	7	8	0	0	2	Лабораторные работы
6.	Тема 6. Деревья цифрового поиска	7	9	0	0	2	Лабораторные работы

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
7.	Тема 7. Создание одностороннего списка. Создание двунаправленного списка. Добавление нового узла в начало одностороннего списка. Добавление нового узла в начало двунаправленного списка. Добавление нового узла в конец одностороннего списка. Добавление нового узла в конец двунаправленного списка. Добавление нового узла в определенное место одностороннего списка. Добавление нового узла в определенное место двунаправленного списка. Удаление первого узла одностороннего списка. Удаление первого узла двунаправленного списка. Удаление последнего узла одностороннего списка. Удаление последнего узла двунаправленного списка. Удаление заданного узла одностороннего списка. Удаление заданного узла двунаправленного списка.	7	10-13	0	0	8	Лабораторные работы
8.	Тема 8. Программирование в среде Visual Studio. Создание приложений с графическим интерфейсом	7	14-18	0	0	10	Лабораторные работы
.	Тема . Итоговая форма контроля	7		0	0	0	Зачет

N	Раздел Дисциплины/ Модуля	Семестр	Неделя семестра	Виды и часы аудиторной работы, их трудоемкость (в часах)			Текущие формы контроля
				Лекции	Практические занятия	Лабораторные работы	
	Итого			0	0	36	

## 4.2 Содержание дисциплины

### Тема 1. Структуры данных на языках программирования. Внутренняя и внешняя сортировки

#### лабораторная работа (4 часа(ов)):

Под термином сортировка понимается процедура перестановки элементов множества в определённом порядке. Методы сортировки классифицируются на внутренние и внешние. При внутренней сортировке данные размещаются в оперативной памяти. При внешней сортировке данные размещаются во внешней памяти. К внешней сортировке прибегают в случаях, когда невозможно разместить в оперативной памяти все данные.

### Тема 2. Сортировка числового массива методом пузырька. Сортировка массива объектов методом пузырька. Сортировка числового массива методом просеивания. Сортировка числового массива методом Шелла. Сортировка числового массива методом быстрой сортировки.

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

Метод пузырька. Производятся последовательные просмотры массива и каждый раз пару за парой сравниваются соседние числа. Если числа в паре не расположены в порядке возрастания меняем их местами. Затем переходим к следующей паре. Сортировка считается законченной, если в ходе просмотра не была произведена ни одна перестановка. Можно использовать переменную, например, flag, чтобы зафиксировать факт перестановки, ? в начале каждого просмотра переменной присваивается значение 0 (False) и если в ходе просмотра была выполнена хотя бы одна перестановка, значение переменной flag меняется на 1 (True). Таким образом, по значению этой переменной определяем, нужен или нет ещё один просмотр. Метод просеивания Выполняется так же как метод пузырька, но после перестановки элементов величина с меньшим значением передвигается к началу массива, насколько это возможно. Она сравнивается в обратном порядке со всеми предшествующими элементами массива. Если значение меньше, чем у предшествующего элемента массива, то выполняется перестановка. Как только встречается элемент с меньшим значением, то процесс продвижения к началу массива прекращается и нисходящее сравнение возобновляется с той же позиции, с которой начался обратный ход. void sort\_sift(int \*x, int dim, int (\*sravn)(int, int)) { int i,j,t; for (i=0;i<dim;i++){ t=x[i]; for (j=i-1;(j>=0) && (sravn(x[j],t)>0); j--) x[j+1]=x[j]; x[j+1]=t; } } Метод Шелла Метод предложен в 1959 году, автор ? Donald Shell. Так же как и метод просеивания, состоит из прямого и обратного хода. Но сравниваются и обмениваются не непосредственные соседи, а элементы, отстоящие на заданном расстоянии. Когда обнаружена перестановка, цепочка вторичных сравнений охватывает те элементы, которые входили в последовательность первичных просмотров. void sort\_shell(int \*x, int dim, int (\*sravn)(int, int)) { int i,j,d,t; d=dim/2; while (d>=1){ for (i=d;i<dim;i++){ t=x[i]; for (j=i-d;(j>=0) && (sravn(x[j],t)>0); j-=d) x[j+d]=x[j]; x[j+d]=t; } d= d/2; } } Быстрая сортировка 1. Выбираем в массиве (случайным образом) какой-нибудь элемент . 2. Просматриваем массив, двигаясь слева направо, пока не найдем элемент . 3. Просматриваем список, двигаясь справо налево, пока не найдем элемент . 4. Меняем местами элементы и . 5. Продолжим процесс просмотра, пока два просмотра не встретятся. В результате массив разделится на две части: левую с ключами, меньшими чем , и правую ? с ключами, большими . 6. Применяем приведённую процедуру для каждой из полученных частей.

### Тема 3. Метод естественного слияния.

#### лабораторная работа (2 часа(ов)):

Сортировка слиянием Как и quickSort, но, вместо разделения по опорному элементу массив просто делится пополам. Рекурсивный алгоритм обходит получившееся дерево слияния в прямом порядке. Каждый уровень представляет собой проход сортировки слияния - операцию, полностью переписывающую массив. Один из способов состоит в слиянии двух упорядоченных последовательностей при помощи вспомогательного буфера, равного по размеру общему количеству имеющихся в них элементов. Элементы последовательностей будут перемещаться в этот буфер по одному за шаг. Результатом является упорядоченная последовательность, находящаяся в буфере. Каждая операция слияния требует  $n$  пересылок и  $n$  сравнений, где  $n$  - общее число элементов.

#### **Тема 4. Алгоритмы внешней сортировки.**

##### **лабораторная работа (2 часа(ов)):**

Если сортируемый файл настолько велик, что не помещается в оперативной памяти компьютера, то прибегают к методам сортировки, для которых используют термин внешняя сортировка (external sorting). Абстрактная модель сортировки построена на предположении, что сортируемый файл не помещается в оперативной памяти компьютера. В распоряжении имеется:  $N$  записей на внешнем устройстве, сортировку которых необходимо выполнить; объем оперативной памяти, достаточный для хранения  $M$  записей;  $2P$  внешних устройств, которыми можно пользоваться во время сортировки. Большинство известных методов внешней сортировки основаны на принципах распределения и слияния. Во время первого прохода по файлу производится его разбиение на блоки (размером меньшим, чем объем оперативной памяти) и выполняются операции сортировки этих блоков. Затем отсортированные блоки сливают в новый файл. Метод естественного слияния Метод естественного слияния основывается на распознавании серий, их распределении и последующем слиянии. Сортировка выполняется за несколько шагов, в каждом из которых выполняются: ? распределение файла А по файлам В и С ? слияние файлов В и С в файл А. При распределении распознается первая серия элементов и переписывается в файл В, вторая серия записывается в файл С, следующая серия снова в В и т.д. При слиянии первая серия записей файла В сливается с первой серией файла С, вторая серия В со второй серией С и т.д. Если просмотр одного файла заканчивается раньше, чем просмотр другого (по причине разного числа серий), то остаток недопросмотренного файла целиком копируется в конец файла А. Процесс завершается, когда в файле А остается только одна серия.

#### **Тема 5. Алгоритмы поиска**

##### **лабораторная работа (2 часа(ов)):**

Алгоритм бинарного поиска.

#### **Тема 6. Деревья цифрового поиска**

##### **лабораторная работа (2 часа(ов)):**

Алгоритмы построения деревьев поиска.

#### **Тема 7. Создание одностороннего списка. Создание двунаправленного списка.**

**Добавление нового узла в начало одностороннего списка. Добавление нового узла в начало двунаправленного списка. Добавление нового узла в конец одностороннего списка. Добавление нового узла в конец двунаправленного списка.**

**Добавление нового узла в определенное место одностороннего списка. Добавление нового узла в определенное место двунаправленного списка. Удаление первого узла одностороннего списка. Удаление первого узла двунаправленного списка.**

**Удаление последнего узла одностороннего списка. Удаление последнего узла двунаправленного списка. Удаление заданного узла одностороннего списка.**

**Удаление заданного узла двунаправленного списка.**

##### **лабораторная работа (8 часа(ов)):**

Линейный список - это конечная последовательность однотипных элементов (узлов), возможно, с повторениями. Количество элементов в последовательности называется длиной списка, причем длина в процессе работы программы может изменяться. При работе со списками чаще всего приходится выполнять следующие операции: ? найти элемент с заданным свойством; ? вставить дополнительный элемент до или после указанного узла; ? исключить определенный элемент из списка; ? упорядочить узлы линейного списка в определенном порядке. Методы хранения линейных списков разделяются на методы последовательного и связанного хранения. При последовательном хранении элементы линейного списка размещаются в массиве фиксированных размеров. Например, int a[10]; double z[20]; Элемент (узел) списка Элемент списка записывается в виде структуры struct ListInt{ int x; ListInt \*next; }; или так typedef struct ListInt \*Link; struct ListInt{ int x; Link next; }; Элемент (узел) двунаправленного списка Элемент списка записывается в виде структуры struct DListInt{ int x; DListInt \*last; DListInt \*next; };

## Тема 8. Программирование в среде Visual Studio. Создание приложений с графическим интерфейсом

### лабораторная работа (10 часа(ов)):

Описание и применение среды программирования.

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

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
1.	Тема 1. Структуры данных на языках программирования. Внутренняя и внешняя сортировки	7	1-2		4	Лабораторные работы
2.	Тема 2. Сортировка числового массива методом пузырька. Сортировка массива объектов методом пузырька. Сортировка числового массива методом просеивания. Сортировка числового массива методом Шелла. Сортировка числового массива методом быстрой сортировки.	7	3-5		6	Лабораторные работы
3.	Тема 3. Метод естественного слияния.	7	6		2	Лабораторные работы
4.	Тема 4. Алгоритмы внешней сортировки.	7	7		2	Лабораторные работы
5.	Тема 5. Алгоритмы поиска	7	8		2	Лабораторные работы
6.	Тема 6. Деревья цифрового поиска	7	9		2	Лабораторные работы

N	Раздел Дисциплины	Семестр	Неделя семестра	Виды самостоятельной работы студентов	Трудоемкость (в часах)	Формы контроля самостоятельной работы
7.	Тема 7. Создание однонаправленного списка. Создание двунаправленного списка. Добавление нового узла в начало однонаправленного списка. Добавление нового узла в начало двунаправленного списка. Добавление нового узла в конец однонаправленного списка. Добавление нового узла в конец двунаправленного списка. Добавление нового узла в определенное место однонаправленного списка. Добавление нового узла в определенное место двунаправленного списка. Удаление первого узла однонаправленного списка. Удаление первого узла двунаправленного списка. Удаление последнего узла однонаправленного списка. Удаление последнего узла двунаправленного списка. Удаление заданного узла однонаправленного списка. Удаление заданного узла двунаправленного списка.	7	10-13		8	Лабораторные работы
8.	Тема 8. Программирование в среде Visual Studio. Создание приложений с графическим интерфейсом	7	14-18		10	Лабораторные работы
	Итого				36	

## 5. Образовательные технологии, включая интерактивные формы обучения

Применяются компьютерные интерактивные технологии, виртуальная аудитория.

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

### **Тема 1. Структуры данных на языках программирования. Внутренняя и внешняя сортировки**

Лабораторные работы , примерные вопросы:

Лабораторная работа номер 1 Тема: сортировка данных. Цель работы: реализовать необходимый алгоритм сортировки данных для решения поставленной задачи, соответствующей заданному варианту. Содержание работы: реализовать консольное приложение на языке программирования высокого уровня C++, которое на определённые входные файлы будет давать корректные выходные файлы, укладываясь в ограничения по памяти и по времени работы. Результат лабораторной работы: реализованный алгоритм на языке высокого уровня C++.

### **Тема 2. Сортировка числового массива методом пузырька. Сортировка массива объектов методом пузырька. Сортировка числового массива методом просеивания. Сортировка числового массива методом Шелла. Сортировка числового массива методом быстрой сортировки.**

Лабораторные работы , примерные вопросы:

Лабораторная работа номер 2 Тема: поиск. Цель работы: реализовать необходимый алгоритм поиска данных для решения поставленной задачи, соответствующей заданному варианту. Содержание работы: реализовать консольное приложение на языке программирования высокого уровня C++, которое на определённые входные файлы будет давать корректные выходные файлы, укладываясь в ограничения по памяти и по времени работы. Результат лабораторной работы: реализованный алгоритм на языке высокого уровня C++. Для выполнения лабораторной работы номер 2 студенту необходимо изучить следующие алгоритмы поиска: ┌ линейный поиск ┌ бинарный поиск ┌ поиск по методу золотого сечения

### **Тема 3. Метод естественного слияния.**

Лабораторные работы , примерные вопросы:

Лабораторная работа номер 3 Тема: длинная арифметика. Цель работы: реализовать необходимый алгоритм длинной арифметики для решения поставленной задачи, соответствующей заданному варианту. Содержание работы: реализовать консольное приложение на языке программирования высокого уровня C++, которое на определённые входные файлы будет давать корректные выходные файлы, укладываясь в ограничения по памяти и по времени работы. Результат лабораторной работы: реализованный алгоритм на языке высокого уровня C++. Для выполнения лабораторной работы номер 3 студенту необходимо изучить следующие алгоритмы для работы с длинными числами: ┌ внутреннее представление длинных чисел ┌ преобразование длинных чисел ┌ сложение длинных чисел ┌ умножение длинных чисел ┌ деление длинных чисел ┌ сравнение длинных чисел

### **Тема 4. Алгоритмы внешней сортировки.**

Лабораторные работы , примерные вопросы:

Лабораторная работа номер 4 Тема: динамические структуры данных. Цель работы: реализовать необходимый алгоритм над списками для решения поставленной задачи, соответствующей заданному варианту. Содержание работы: реализовать консольное приложение на языке программирования высокого уровня C++, которое на определённые входные файлы будет давать корректные выходные файлы, укладываясь в ограничения по памяти и по времени работы. Результат лабораторной работы: реализованный алгоритм на языке высокого уровня C++. Для выполнения лабораторной работы номер 4 студенту необходимо изучить следующие алгоритмы работы со списками:  $\square$  добавление элемента  $\square$  удаление элемента  $\square$  поиск элемента  $\square$  перестановка элементов местами Лабораторная работа номер 5 Тема: бинарные деревья. Цель работы: реализовать необходимый алгоритм обработки бинарных деревьев для решения поставленной задачи, соответствующей заданному варианту. Содержание работы: реализовать консольное приложение на языке программирования высокого уровня C++, которое на определённые входные файлы будет давать корректные выходные файлы, укладываясь в ограничения по памяти и по времени работы. Результат лабораторной работы: реализованный алгоритм на языке высокого уровня C++. Для выполнения лабораторной работы номер 5 студенту необходимо изучить следующие алгоритмы работы с бинарными деревьями:  $\square$  построение сбалансированного дерева  $\square$  построение частично сбалансированного поиска  $\square$  построение дерева поиска  $\square$  построение частично сбалансированного дерева поиска

### **Тема 5. Алгоритмы поиска**

Лабораторные работы , примерные вопросы:

Реализация и визуализация работ 1-4

### **Тема 6. Деревья цифрового поиска**

Лабораторные работы , примерные вопросы:

Реализация и визуализация работ 1-4

**Тема 7. Создание одностороннего списка. Создание двунаправленного списка.**  
**Добавление нового узла в начало одностороннего списка. Добавление нового узла в начало двунаправленного списка. Добавление нового узла в конец одностороннего списка. Добавление нового узла в конец двунаправленного списка. Добавление нового узла в определенное место одностороннего списка. Добавление нового узла в определенное место двунаправленного списка. Удаление первого узла одностороннего списка. Удаление первого узла двунаправленного списка. Удаление последнего узла одностороннего списка. Удаление последнего узла двунаправленного списка. Удаление заданного узла одностороннего списка.**  
**Удаление заданного узла двунаправленного списка.**

Лабораторные работы , примерные вопросы:

Реализация и визуализация работ 1-4

### **Тема 8. Программирование в среде Visual Studio. Создание приложений с графическим интерфейсом**

Лабораторные работы , примерные вопросы:

Реализация и визуализация работ 1-4

### **Итоговая форма контроля**

зачет (в 7 семестре)

Примерные вопросы к зачету:

Программа зачета:

Структуры данных. Внутренняя и внешняя сортировки

Сортировка числового массива методом пузырька. Сортировка массива объектов методом пузырька. Сортировка числового массива методом просеивания. Сортировка числового массива методом Шелла. Сортировка числового массива методом быстрой сортировки.

Метод естественного слияния.

Алгоритмы внешней сортировки.

Алгоритмы поиска

## Деревья цифрового поиска

Создание одностороннего списка. Создание двунаправленного списка. Добавление нового узла в начало одностороннего списка. Добавление нового узла в начало двунаправленного списка. Добавление нового узла в конец одностороннего списка. Добавление нового узла в конец двунаправленного списка. Добавление нового узла в определенное место одностороннего списка. Добавление нового узла в определенное место двунаправленного списка. Удаление первого узла одностороннего списка. Удаление первого узла двунаправленного списка. Удаление последнего узла одностороннего списка. Удаление последнего узла двунаправленного списка. Удаление заданного узла одностороннего списка. Удаление заданного узла двунаправленного списка.

### 7.1. Основная литература:

1. Программирование графики на C++. Теория и примеры : учеб. пособие / В.И. Корнеев, Л.Г. Гагарина, М.В. Корнеева. ? М. : ИД 'ФОРУМ' : ИНФРА-М, 2017. ? 517 с. + Доп. материалы [Электронный ресурс; Режим доступа <http://www.znanium.com>]. ? (Высшее образование). ? [www.dx.doi.org/10.12737/23113](http://www.dx.doi.org/10.12737/23113). <http://znanium.com/catalog/product/562914>
2. Объектно-ориентированное программирование с примерами на C#: Учебное пособие / Хорев П.Б. - М.: Форум, НИЦ ИНФРА-М, 2016. - 200 с.: 70x100 1/16. - (Высшее образование: Бакалавриат) (Обложка) ISBN 978-5-00091-144-0 <http://znanium.com/catalog/product/529350>

### 7.2. Дополнительная литература:

1. Основы алгоритмизации и программирования: Учебное пособие / В.Д. Колдаев; Под ред. Л.Г. Гагариной. - М.: ИД ФОРУМ: ИНФРА-М, 2015. - 416 с.: ил.; 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-8199-0279-0 - Режим доступа: <http://znanium.com/catalog/product/484837>
2. Программирование на C++ с погружением: практические задания и примеры кода - М.:НИЦ ИНФРА-М, 2016. - 80 с.: 60x90 1/16. <http://znanium.com/catalog/product/563294>

### 7.3. Интернет-ресурсы:

Комбинаторные алгоритмы для программистов - <http://www.intuit.ru/studies/courses/65/65/info>  
CyberForum.ru - форум программистов и сисадминов - <http://www.cyberforum.ru/>  
Visual C++ - <http://msdn.microsoft.com/ru-ru/library/60k1461a.aspx>  
Основы объектно-ориентированного программирования -  
<http://www.intuit.ru/studies/courses/71/71/info>  
Основы программирования на языке С - <http://www.intuit.ru/studies/courses/43/43/info>

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

Освоение дисциплины "Практикум по решению задач на компьютере" предполагает использование следующего материально-технического обеспечения:

Мультимедийная аудитория, вместимостью более 60 человек. Мультимедийная аудитория состоит из интегрированных инженерных систем с единой системой управления, оснащенная современными средствами воспроизведения и визуализации любой видео и аудио информации, получения и передачи электронных документов. Типовая комплектация мультимедийной аудитории состоит из: мультимедийного проектора, автоматизированного проекционного экрана, акустической системы, а также интерактивной трибуны преподавателя, включающей тач-скрин монитор с диагональю не менее 22 дюймов, персональный компьютер (с техническими характеристиками не ниже Intel Core i3-2100, DDR3 4096Mb, 500Gb), конференц-микрофон, беспроводной микрофон, блок управления оборудованием, интерфейсы подключения: USB, audio, HDMI. Интерактивная трибуна преподавателя является ключевым элементом управления, объединяющим все устройства в единую систему, и служит полноценным рабочим местом преподавателя. Преподаватель имеет возможность легко управлять всей системой, не отходя от трибуны, что позволяет проводить лекции, практические занятия, презентации, вебинары, конференции и другие виды аудиторной нагрузки обучающихся в удобной и доступной для них форме с применением современных интерактивных средств обучения, в том числе с использованием в процессе обучения всех корпоративных ресурсов. Мультимедийная аудитория также оснащена широкополосным доступом в сеть интернет. Компьютерное оборудование имеет соответствующее лицензионное программное обеспечение.

Компьютерный класс, представляющий собой рабочее место преподавателя и не менее 15 рабочих мест студентов, включающих компьютерный стол, стул, персональный компьютер, лицензионное программное обеспечение. Каждый компьютер имеет широкополосный доступ в сеть Интернет. Все компьютеры подключены к корпоративной компьютерной сети КФУ и находятся в едином домене.

Необходим доступ к виртуальной аудитории.

Программа составлена в соответствии с требованиями ФГОС ВПО и учебным планом по направлению 44.03.05 "Педагогическое образование (с двумя профилями подготовки)" и профилю подготовки Математика, информатика и информационные технологии .

Автор(ы):

Галимьянов А.Ф. \_\_\_\_\_

Першагин М.Ю. \_\_\_\_\_

"\_\_" 201\_\_ г.

Рецензент(ы):

Авхадиев Ф.Г. \_\_\_\_\_

"\_\_" 201\_\_ г.