



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Дальневосточный федеральный университет»

(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК ДВФУ

«СОГЛАСОВАНО»

Руководитель ОП
д.ф.-м.н., профессор, академик РАН, Гузев М.А.

(подпись) (Ф.И.О. рук. ОП)
«23» июня 2017 г.

«УТВЕРЖДАЮ»

Заведующая (ий) кафедрой
информатики, математического и компьютерного
моделирования
(название кафедры)



Чеботарев А.Ю.
(подпись) (Ф.И.О. зав. каф.)
«23» июня 2017 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
Основы информатики и программирования
Направление подготовки 09.03.03 Прикладная информатика
профиль «Прикладная информатика в компьютерном дизайне»
Форма подготовки очная

курс 1, семестр 1,2
лекции 54 час.
практические занятия 72 час.
лабораторные работы 54 час.
в том числе с использованием МАО лек.20/ пр. 20, лаб. 50
всего часов аудиторной нагрузки 180 час.
в том числе с использованием МАО 90
самостоятельная работа 216 час.
на подготовку к экзамену 72 час.
контрольные работы (количество) _____
курсовая работа/курсовой проект _____ семестр
зачет семестры
экзамен 1,2 семестр

Рабочая программа составлена в соответствии с требованиями образовательного стандарта, самостоятельно установленного ДВФУ, принятого решением Ученого совета Дальневосточного федерального университета, протокол от 28.01.2016 № 01-16, и введенного в действие приказом ректора ДВФУ от 18.02.2016 № 12-13-235. Рабочая программа обсуждена на заседании кафедры информатики, математического и компьютерного моделирования, протокол № 22 от «23» июня 2017 г.
Заведующий кафедрой профессор Чеботарев А.Ю.
Составитель: доцент Кленина Н. В.,

Оборотная сторона титульного листа РПУД

I. Рабочая программа пересмотрена на заседании кафедры:

Протокол от « _____ » _____ 201 г. № _____

Заведующий кафедрой _____
(подпись) (А.С. Чеботарев)

II. Рабочая программа пересмотрена на заседании кафедры:

Протокол от « _____ » _____ 200 г. № _____

Заведующий кафедрой _____
(подпись) (А.С. Чеботарев)

АННОТАЦИЯ

Рабочая программа дисциплины «Основы информатики и программирования» разработана для студентов профиля «Прикладная информатика в компьютерном дизайне» по направлению 09.03.03 «Прикладная информатика».

Общая трудоемкость освоения дисциплины составляет 13 зачетных единиц. Учебным планом предусмотрены, лабораторные работы (54 часа), практические занятия (72 часа), лекции (54 часа) самостоятельная работа студента (216 часов). Дисциплина реализуется на 1 курсе в 1-2 семестрах.

Место дисциплины «Основы информатики и программирования» в структуре ООП ВПО

Дисциплина «Основы информатики и программирования» относится к циклу профессиональных дисциплин ОП (базовая часть Математического и естественнонаучного цикла).

Дисциплина «Основы информатики и программирования» логически и содержательно связана с такими курсами, как «Вычислительные системы, сети и телекоммуникации», «Проектирование ИС», «Технология программирования», «Операционные системы», «Программная инженерия».

Логическая взаимосвязь

с Гуманитарным, социальным и экономическим циклом в разделах: проблема истины; действительность, мышление, логика и язык; структура научного познания, его методы и формы; сущность, формы, функции исторического знания.

Содержательно-методическая взаимосвязь с циклами:

Информационные ресурсы и системы, в разделах: назначение и виды ИКТ; Информационная безопасность; технологии сбора, накопления, обработки, передачи и распространения информации; модели данных;

Программно-технические средства, в разделах: физические основы компьютерной техники и средств передачи информации, принципы работы технических устройств, процессов функционирования вычислительных систем.

Близкая по содержанию дисциплина – «Программная инженерия».

Предшествующая дисциплина: «Информатика и ИКТ», школьный курс,

Теоретические дисциплины и практики, для которых освоение данной дисциплины необходимо как предшествующее: «Проектирование ИС», «Вычислительные системы, сети и телекоммуникации», «Операционные системы».

Содержание дисциплины охватывает знания о теоретических основах информатики и программирования, включая основные понятия информатики, теорию кодирования информации, базовые методы алгоритмизации, динамические структуры данных, высокоуровневые языки и среды программирования, технические средства информатики, а также умения выполнять различные операции преобразования информации, анализ информации, реализовывать алгоритмы в различных средах программирования.

Уделяется внимание глубокому изучению практических аспектов применения стандартных динамических структур данных, методов алгоритмизации.

Цель изучения дисциплины «Основы информатики и программирования» – овладение основами информационных технологий, получение знаний об основах алгоритмизации, формирование начальных умений формализации и моделирования информации, формирование научного мировоззрения, развитие логического и алгоритмического мышления..

Задачи:

– создание необходимой основы для использования современных средств вычислительной техники и прикладных программ при изучении студентами естественнонаучных, общепрофессиональных и специальных дисциплин;

– освоение предусмотренного программой теоретического материала и приобретение практических навыков использования информационных систем и технологий на базе современных ПК и навыков программирования.

– достижение понимания студентами сущности и проблем развития современного информационного общества, понимание сущности и значения информации в развитии современного информационного общества, осознание опасностей и угроз, возникающих в этом процессе;

– освоение студентами основных понятий информатики;

– выработка умений: использовать, обобщать и анализировать информацию, ставить цели и находить пути их достижения в условиях формирования и развития информационного общества; применять системный подход и математические методы в формализации решения прикладных задач;

– формирование навыков применения к решению прикладных задач базовых алгоритмов обработки информации, выполнения оценки сложности алгоритмов, программировать и тестировать программы

– формирование и развитие способностей к суждениям, способностей логически верно, аргументированно и ясно строить устную и письменную речь; самостоятельно приобретать и использовать в практической деятельности новые знания и умения; стремления к саморазвитию;

Требования к «входным» знаниям, приобретенным в результате освоения предшествующих дисциплин:

- владение основами информационных технологий;
- знание основ алгоритмизации;
- начальные умения формализации и моделирования.

Для успешного изучения дисциплины «Информатика и программирование» у обучающихся должны быть сформированы следующие предварительные компетенции в области информатики, информационных технологий и программирования.

Для изучения дисциплины студент должен:

Знать:

- простейшие базовые алгоритмы обработки информации, выполнять оценку сложности алгоритмов, программировать и тестировать программы
- средства работы с информацией в глобальных компьютерных сетях
- базовые инструменты проектирования и структурирования программных продуктов
- обладать высокой мотивацией к выполнению профессиональной деятельности

Уметь:

- ставить и решать прикладные задачи с использованием современных информационно-коммуникационных технологий
- программировать на одном из алгоритмических языков;
- строить простые оконные приложения;
- решать простые задачи на алгоритмизацию.

Владеть:

- способностью самостоятельно приобретать и использовать в практической деятельности новые знания и умения, стремиться к саморазвитию
- способностью осознавать социальную значимость своей будущей профессии, обладать высокой мотивацией к выполнению профессиональной деятельности

Изучаемая дисциплина формирует основные компетенции специалиста в области информатики, информационных технологий и программирования.

В результате освоения данной дисциплины бакалавр приобретает знания, умения и навыки, обеспечивающие достижение целей основной образовательной программы «Прикладная информатика».

Дисциплина должна:

- познакомить студентов с теоретическими основами информатики;

- научить студентов самостоятельно осваивать дополнительные инструментальные средства программирования;
- научить студентов использовать широко распространенные алгоритмы и алгоритмические методы;
- научить студентов разрабатывать прикладные программы, использующие языки программирования высокого уровня.

В результате изучения данной дисциплины у обучающихся формируются следующие общекультурные/ общепрофессиональные/ профессиональные компетенции (элементы компетенций).

Код и формулировка компетенции	Этапы формирования компетенции	
ОК-3 Способность проявлять инициативу и принимать ответственные решения, осознавая ответственность за результаты своей профессиональной деятельности	Знает	Понятие и назначение формализации. Математические методы в формализации задачи.
	Умеет	Применять математические (численные) методы при решении задач.
	Владеет	Навыками научно-исследовательской деятельности: системному подходу к исследуемой проблеме по поставленному заданию. Формализации постановки задачи.
ПК-2 Способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение	Знает	Основные этапы постановки задачи. Набор инструментальных средств, достаточный для решения задач средней сложности.
	Умеет	Формировать неформальную и формальную постановку задачи. Обосновывать необходимость решения задачи.
	Владеет	.Навыками разработки и адаптации программного обеспечения средней сложности.
ПК-13 Способностью проводить тестирование компонентов программного обеспечения	Знает	Основные этапы коллективной разработки программного продукта, технологию тестирования программного продукта.
	Умеет	Обеспечивать формировать наборы тестов для проверки программного продукта.

	Владеет	Навыками работы в интегрированной среде; методами алгоритмизации и программирования; навыками тестирования и отладки приложений.
ПК-15 Способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач	Знает	Основы теории баз данных.
	Умеет	Использовать основные законы и требования к разработке и ведению баз данных в профессиональной деятельности.
	Владеет	Навыками проектирования и реализации баз данных при решении прикладных задач.

Для формирования вышеуказанных компетенций в рамках дисциплины «Основы информатики и программирования» применяются следующие методы активного/ интерактивного обучения: лекция-беседа, лекция-дискуссия, метод группового обучения, метод автоматизированного обучения.

При выполнении различных видов работ используются следующие технологии:

1. *Работа в команде* – совместная деятельность обучающихся в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности.
2. *Проблемное обучение* – стимулирование обучающихся к самостоятельному приобретению знаний, необходимых для решения конкретной проблемы.
3. *Контекстное обучение* – мотивация студентов магистратуры к усвоению знаний путём выявления связей между конкретным знанием и его применением.
4. *Обучение на основе опыта* – активизация познавательной деятельности студентов бакалавриата за счёт ассоциации и собственного опыта с предметом обучения.

I. СТРУКТУРА И СОДЕРЖАНИЕ ТЕОРЕТИЧЕСКОЙ ЧАСТИ КУРСА

Раздел 1. Основные понятия и принципы информатики (34/0 час.)

Тема 1.1. Предмет информатики. (4/0 час.)

- Тенденции и проблемы развития современного информационного общества. История развития информатики. Этапы развития информационных технологий.

- Значение информации в развитии современного информационного общества. Информационная деятельность. Промышленность обработки данных.

- Понятие информации. Место информации в мире. Свойства информации.

Тема 1.2. Основные понятия информатики (4/0 час.)

- Носитель информации: история развития искусственных носителей информации. Характеристики носителей информации, их достоинства и недостатки. Тенденции развития носителей информации

- Понятие сигнала. Состояние носителя информации. Интерпретация сигналов. Набор сигналов.

- Формы информации. Аналоговая информация. Виды аналоговой информации. Понятие дискретизации. Виды дискретной информации.

- Информационное сообщение. Информационные процессы.

- Этапы восприятия информации.

Тема 1.3. Принципы кодирования информации (10/0 час.)

- Общие понятия и правила кодирования информации. Таблицы кодировки. Стандартные таблицы кодировки.

- Принципы кодирования текстовой информации. Принципы кодирования графической информации. Принципы кодирования графической информации. Принципы кодирования видео- информации. Принципы кодирования числовой информации.

Тема 1.4. Двоичное кодирование информации (6/0 час.)

- Машиночитаемые носители информации. Достоинства и недостатки. Виды, устройство и характеристики машиночитаемых носителей. Особенности машиночитаемого кодирования текстовой, графической. видео-, числовой информации.

- Информационный объем сообщения. Единицы измерения информации. Способы сокращения информационного объема.

Тема 1.5. Кодирование рассуждений (8/0 час.)

- Определение понятия высказывание. Кодирование высказываний. Логическое выражение.

- Определение алгоритма. Свойства алгоритма. Способы кодирования алгоритма. Исполнители алгоритма.

- Автоматический исполнитель алгоритма. Техническое устройство, характеристики. Законы исполнения алгоритма.

Тема 1.6. Опасности и угрозы информационного общества. Основы защиты информации. (2/0 час.)

- Программное обеспечение. Виды, тенденции развития ПО..

- Вредоносное программное обеспечение. Виды вредоносного программного обеспечения. Основы защиты информации.

Раздел 2. Теоретические основы программирования. (20/0 час.)

Тема 2.1. Типы данных. (8/0 час.)

Внутренний формат. Диапазон значений. Набор операций.

Тема 2.2. Базовые положения программирования. (12/0 час.)

- Потоки управления данными в программах.
- Структура программы.
- Стил программирования.
- Этапы жизненного цикла программ.

II. СТРУКТУРА И СОДЕРЖАНИЕ ПРАКТИЧЕСКОЙ ЧАСТИ КУРСА

Практические занятия (72 час.)

Занятие 1. Основы языка программирования (8/0 час.)

- Понятие синтаксиса и семантики ЯП.
- Основные конструкции ЯП.
- Характерная для описания ЯП терминология.
- Операторы ЯП.

Занятие 2. Основы программирования (10/0 час.)

- Линейные алгоритмы.

- Алгоритмы с ветвлением.
- Циклические алгоритмы.
- Базовые алгоритмы.

Занятие 3. Структуры данных (18/0 час.)

- Понятие структур данных. Статические структуры данных. Динамические структуры данных.
- Средства динамического выделения памяти, освобождения памяти, обращения к памяти.

Занятие 4. Динамическая структура данных список. (18/0 час.)

- Динамическая структура данных линейный список: определение, схема, виды. Набор операций над линейным списком.
- Алгоритмы с использованием линейного списка. Длинная арифметика. Менеджер памяти.

Занятие 5. Стандартные динамические структуры данных. (18/0 час.)

- Динамическая структура данных стек: определение, схема, способы реализации стека. Набор операций над стеком.
- Алгоритмы с использованием стека. Анализ выражения. Вычисление выражений. Преобразование выражений.
- Динамическая структура данных очередь: определение, схема. Набор операций над очередью.
- Алгоритмы с использованием очереди.
- Динамическая структура данных дерево: определение, схема. Набор операций над деревом. Алгоритмы сортировки с использованием дерева.

Лабораторные работы (54 час./0)

Лабораторная работа №1. Простые алгоритмы (10/0 час.)

Реализация простых алгоритмов с массивами. Основы коллективного взаимодействия при разработке приложения. Формализация задачи и оценка сложности алгоритмов. Анализ надежности алгоритмов.

Лабораторная работа №2. Рекурсия (10/0 час.)

Аспекты реализации рекурсии. Использование стека при реализации рекурсии. Использование рекурсии при реализации алгоритмов сортировки и поиска. Производительность рекурсивных алгоритмов.

- Примеры рекурсивных алгоритмов.
- Классические рекурсивные алгоритмы.
- Рекурсивные алгоритмы на динамических структурах данных.

Лабораторная работа №3. Алгоритмы сортировки (14/0 час.)

Основные параметры и классификация алгоритмов сортировки.

- Сортировка чисел и строк.
- Эффективность алгоритмов сортировки.
- Оценка сложности алгоритмов сортировки
- Использование динамических сортировок.
- Рекурсивные алгоритмы сортировки.

Лабораторная работа №4. Алгоритмы поиска (10/0 час.)

Основные параметры и классификация алгоритмов сортировки.

- Сортировка чисел и строк.
- Эффективность алгоритмов сортировки.
- Оценка сложности алгоритмов сортировки
- Использование динамических сортировок.
- Рекурсивные алгоритмы сортировки.

Лабораторная работа №5. Оконные приложения (10/ час.)

- Среда программирования Lazarus.
- Запуск и отладка программ в среде Lazarus (консольное приложение).
- Оконное приложение. Компоненты. Свойства компонент. События.
- Средства создания и редактирования форм.
- Программирование оконных приложений.

III. УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ.

Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине «Основы информатики и программирования» представлено в Приложении 1 и включает в себя:

план-график выполнения самостоятельной работы по дисциплине, в том числе примерные нормы времени на выполнение по каждому заданию;

характеристика заданий для самостоятельной работы обучающихся и методические рекомендации по их выполнению;

требования к представлению и оформлению результатов самостоятельной работы;

критерии оценки выполнения самостоятельной работы.

План-график выполнения самостоятельной работы по дисциплине

№ п/п, название	Дата/сроки выполнения	Вид СРС	Примерные нормы времени на выполнение	Форма контроля
1. Стек	Вторая неделя 3 семестра	ИДЗ	1 неделя	Назначение в системе Bb dvfu
2. Очередь	Четвёртая неделя 3 семестра	ИДЗ	1 неделя	Назначение в системе Bb dvfu
3. Сортировка	Восьмая неделя 3 семестра	ИДЗ	3 недели	Назначение в системе Bb dvfu
4. Поиск	Одиннадцатая неделя 3 семестра	ИДЗ	3 неделя	Коллоквиум
5. Выражения	Тринадцатая неделя 3 семестра	ИДЗ	1 неделя	Назначение в системе Bb dvfu
6. Калькулятор	Вторая неделя	ИДЗ	2 недели	Назначение в

	4 семестра			системе Bb dvfu
7. Движение объектов	Четвертая неделя 4 семестра	ИДЗ	1 неделя	Тестирование
8. Проект программного продукта	Десятая неделя 4 семестра	ИДЗ	3 недели	Письменный отчет
9. Реализация программного продукта в оконной среде	Пятнадцатая неделя 4 семестра	ИДЗ	3 недели	Тестирование
10. Презентация программного продукта	Зачётная неделя 4 семестра	ИДЗ	1 час	Собеседование

Материалы для самостоятельной работы студентов подготовлены в виде индивидуальных домашних заданий по каждой теме (образцы типовых ИДЗ представлены в разделе «Материалы для самостоятельной работы студентов»). Работа должна быть отправлена преподавателю на проверку в системе CATS либо по электронной почте по соответствующему адресу. Оформление: исходный или исполняемый файл в зависимости от требований задания. Критерии оценки: студент получает максимальный балл, если работа выполнена без ошибок и оформлена в соответствии с требованиями преподавателя.

ХАРАКТЕРИСТИКА ЗАДАНИЙ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Ознакомление с основными концепциями динамического распределения памяти
2. Знакомство с научной и научно-популярной литературой по методам сортировки и поиска информации.
3. Знакомство с периодическими изданиями по теории алгоритмизации.

4. Изучение статей по дисциплине.
5. Знакомство с приложениями алгоритмов сортировки и поиска современных информационных технологиях.

ТРЕБОВАНИЯ К ПРЕДСТАВЛЕНИЮ И ОФОРМЛЕНИЮ РЕЗУЛЬТАТОВ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Устный доклад по указанной теме.
2. Презентация к докладу по указанной теме.
3. Демонстрация программного кода (мультимедийный экран), подтверждение работоспособности программы.

Дополнительный вариант:

Реферат по указанной теме.

КРИТЕРИИ ОЦЕНКИ ВЫПОЛНЕНИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

1. Качество доклада (10 баллов)
2. Качество презентации (10 баллов)
3. Качество программного кода (10 баллов)
4. Качество ответов на вопросы (10 баллов)

IV. КОНТРОЛЬ ДОСТИЖЕНИЯ ЦЕЛЕЙ КУРСА

Изучение дисциплины «Основы информатики и программирования» предусматривает:

- предоставление теоретического материала в соответствии с программой, с использованием электронного раздаточного материала;
- предоставление теоретического материала посредством электронных обучающих программ.
- выполнение домашних заданий;
- выполнение индивидуальных заданий;
- обязательная проработка материала, который будет разбираться на занятии с подбором дополнительных материалов.

Текущий контроль. Предусматривает учет посещения студентами занятий в течение периода обучения и оценку своевременности и качества выполнения студентами тестов и домашних заданий.

Итоговый контроль. Предусматривает рейтинговую оценку по учебной дисциплине в течение семестра и зачет.

№ п/п	Контролируемые разделы / темы дисциплины	Коды и этапы формирования компетенций	Оценочные средства - наименование	
			текущий контроль	промежуточная аттестация
1.	Динамические структуры данных	<p>ОК-3 Способность проявлять инициативу и принимать ответственные решения, осознавая ответственность за результаты своей профессиональной деятельности</p> <p>ПК-2 Способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение</p> <p>ПК-13 Способностью проводить тестирование компонентов программного обеспечения</p> <p>ПК-15 Способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач</p>	<p>1. Решение задач по изучаемой теме на практических занятиях;</p> <p>2. Летучий устный или письменный опрос студентов во время занятия по изучаемому материалу;</p> <p>3. Теоретические диктанты;</p> <p>4. Индивидуальные домашние задания;</p> <p>5. Тесты.</p> <p>6. Вопросы к зачету.</p>	
2.	Классические алгоритмы для динамически структуры данных	<p>ОК-3 Способность проявлять инициативу и принимать ответственные решения, осознавая ответственность за результаты своей профессиональной деятельности</p> <p>ПК-2 Способностью разрабатывать, внедрять и адаптировать прикладное программное</p>	<p>1. Решение задач по изучаемой теме на практических занятиях;</p> <p>2. Летучий устный или письменный опрос студентов во время занятия по изучаемому материалу;</p> <p>3. Теоретические диктанты;</p> <p>4. Индивидуальные домашние задания;</p> <p>5. Тесты.</p> <p>6. Вопросы к зачету.</p>	

		<p>обеспечение ПК-13 Способностью проводить тестирование компонентов программного обеспечения</p> <p>ПК-15 Способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач</p>	
3.	Алгоритмы сортировки данных	<p>ОК-3 Способность проявлять инициативу и принимать ответственные решения, осознавая ответственность за результаты своей профессиональной деятельности</p> <p>ПК-2 Способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение</p> <p>ПК-13 Способностью проводить тестирование компонентов программного обеспечения</p> <p>ПК-15 Способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач</p>	<ol style="list-style-type: none"> 1. Решение задач по изучаемой теме на практических занятиях; 2. Летучий устный или письменный опрос студентов во время занятия по изучаемому материалу; 3. Теоретические диктанты; 4. Индивидуальные домашние задания; 5. Тесты. 6. Вопросы к зачету.
4.	Алгоритмы поиска данных	<p>ОК-3 Способность проявлять инициативу и принимать ответственные решения, осознавая ответственность за результаты своей профессиональной деятельности</p> <p>ПК-2 Способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение</p> <p>ПК-13 Способностью проводить тестирование компонентов программного обеспечения</p>	<ol style="list-style-type: none"> 1. Решение задач по изучаемой теме на практических занятиях; 2. Летучий устный или письменный опрос студентов во время занятия по изучаемому материалу; 3. Теоретические диктанты; 4. Индивидуальные домашние задания; 5. Тесты. 6. Вопросы к зачету.

		ПК-15 Способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач	
5.	Реализация оконных приложений	ОК-3 Способность проявлять инициативу и принимать ответственные решения, осознавая ответственность за результаты своей профессиональной деятельности ПК-2 Способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение ПК-13 Способностью проводить тестирование компонентов программного обеспечения ПК-15 Способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач	1. Решение задач по изучаемой теме на практических занятиях; 2. Летучий устный или письменный опрос студентов во время занятия по изучаемому материалу; 3. Теоретические диктанты; 4. Индивидуальные домашние задания; 5. Тесты. 6. Вопросы к зачету.

Типовые контрольные задания, методические материалы, определяющие процедуры оценивания знаний, умений и навыков и (или) опыта деятельности, а также критерии и показатели, необходимые для оценки знаний, умений, навыков и характеризующие этапы формирования компетенций в процессе освоения образовательной программы, представлены в Приложении 2.

V. СПИСОК УЧЕБНОЙ ЛИТЕРАТУРЫ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Основная литература (электронные и печатные издания)

1. Алгоритмизация и программирование : Учебное пособие / С.А. Канцедал. - М.: ИД ФОРУМ: НИЦ ИНФРА-М, 2013. - 352 с.: ил.; 60x90 1/16. - (Профессиональное образование). (переплет) ISBN 978-5-8199-0355-1

2. Царев, Р.Ю. Информатика и программирование [Электронный ресурс] : учеб. пособие / Р. Ю. Царев, А. Н. Пупков, В. В. Самарин, Е. В. Мыльникова. – Красноярск : Сиб. федер. ун-т, 2014. – 132 с. - ISBN 978-5-7638-3008-8
3. Программирование : учебник для вузов / Г. С. Иванова. 3-е изд., стер. Москва: КноРус , 2014. 426 с
4. Программирование на языке высокого уровня. Программирование на языке Object Pascal: Учебное пособие / Т.И. Немцова; Под ред. Л.Г. Гагариной. - М.: ФОРУМ: ИНФРА-М, 2015. - 496 с.: ил.; 60x90 1/16. - (ПО). (п, cd rom) ISBN 978-5-8199-0372-8, 300 экз.
5. Delphi: программирование в примерах и задачах. Практикум: Учебное пособие / Г.М. Эйдлина, К.А. Милорадов. - М.: ИЦ РИОР: НИЦ Инфра-М, 2012. - 116 с.: 60x88 1/16. - (Высшее образование: Бакалавриат). (обложка) ISBN 978-5-369-01084-6
6. Методы программирования: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, Ю.В. Кулаков, Ю.В. Минин, В.Г. Однолько. - Тамбов: Изд-во ФГБОУ ВПО "ТГТУ", 2012. - 144 с.
7. Бариллов И.В. Информатика и программирование. - Шахты: ЮРГУЭС, 2005. - 12.
8. Серджик Роберт Фундаментальные алгоритмы на С. Алгоритмы на графах. Пер. с англ. / Роберт Серджик. - Спб: «ДиаСофтЮП», 2013 — 480 с.
9. Роганов Е.А. Основы информатики и программирования [Электронный ресурс] / Е.А. Роганов. — 2-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 392 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/73689.html>.
10. Львович И.Я. Основы информатики [Электронный ресурс] : учебное пособие / И.Я. Львович, Ю.П. Преображенский, В.В. Ермолова. — Электрон. текстовые данные. — Воронеж: Воронежский институт

высоких технологий, 2014. — 339 с. — 2227-8397. — Режим доступа:
<http://www.iprbookshop.ru/23359.html>

**Дополнительная литература
(печатные и электронные издания)**

1. Бураков П.В., Косовцева Т.Р. Информатика. Алгоритмы и программирование. Учебное пособие. - СПб НИУ ИТМО, 2013. - 83 с.
<http://window.edu.ru/resource/424/80424>
1. Пратт Т., Зелковиц М. ПТО Языки программирования: разработка и реализация / под общей ред. А. Матросова - СПб: Питер, 2012. - 688 с.
2. teormin.ifmo.ru (csin.ru) — "Теоретический минимум по Информатике". Курсы и учебные материалы, организованные по темам
3. Васильев В.И., Тягунова Т.Н. Культура компьютерного тестирования. Ч.1 Философия адаптивного тестирования. М.: МГУП, 2012.

**Перечень ресурсов информационно-телекоммуникационной сети
«Интернет»**

1. - электронный учебник "Информатика для Вас".
ispu.ru — Информатика. Курс лекций. Жуков В.П., Иваново, 2010г.
(электронное on-line пособие, Ивановский Гос. Энерг. Унив.)
2. <http://www.osu.ru/docs/bachelor/fgos/230700b.pdf/>

Перечень информационных технологий и программного обеспечения

1. Электронная презентация
2. Электронная рассылка
3. Электронный журнал успеваемости
4. Компиляторы с Lazarus и Pascal.
5. ЭУК «Информатика и программирование» в интегрированной платформе электронного обучения Blackboard ДВФУ.
6. Электронные поисковые приложения.

VI. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ

На лабораторных занятиях перед выдачей индивидуальных заданий преподаватель объясняет теоретический материал по заданной теме. Вводит основные требования к его выполнению. Приводит примеры. Необходимо поддерживать непрерывный контакт с аудиторией, отвечать на возникающие у студентов вопросы. На практических занятиях преподаватель разбирает на примерах принципы и аспекты реализации задания по заданной теме.

Во второй части занятия студентам предлагается работать самостоятельно, выполняя задания по теме. Преподаватель контролирует работу студентов, отвечает на возникающие вопросы, подсказывает ход и метод решения. Если знаний полученных в аудитории оказалось недостаточно, студент может самостоятельно повторно просмотреть презентацию, просмотреть практикум с разобранными примерами, которые собраны в изучаемом курсе в системе Вb dvfu.

После выполнения задания, студент отправляет его на проверку преподавателю по электронной почте, либо предъявляет на компьютере во время занятия. Работа должна быть отослана либо одним документом – исходным файлом на языке программирования, либо архивом.

По данному курсу разработаны методические указания, которые выложены в системе Blackboard ДВФУ в соответствующем разделе. Для успешного достижения учебных целей занятий должны выполняться следующие основные требования:

- соответствие действий обучающихся ранее изученным на лекционных и семинарских занятиях методикам и методам.
- максимальное приближение действий студентов к реальным, соответствующим будущим функциональным обязанностям.
- поэтапное формирование умений и навыков, т.е. движение от знаний к умениям и навыкам, от простого к сложному и т.д..

-использование при работе на тренажерах или действующей технике фактических документов, технологических карт, бланков и т.п.

-выработка соответствующих индивидуальных и коллективных умений и навыков.

-распределение времени, отведенного на занятие, на решение каждой задачи;

-подбор иллюстративного материала (графиков, таблиц, схем), необходимого для решения задач, продумывание расположения рисунков и записей на доске или экране.

Студент должен:

-научиться работать с книгой, документацией и схемами, пользоваться справочной и научной литературой.

-научиться работать с электронными литературными источниками.

-формировать умение учиться самостоятельно, т.е. овладевать методами, способами и приемами самообучения, саморазвития и самоконтроля.

Помощь в освоении данного курса окажут электронные учебные курсы:

1. Обучающий курс «Работа со стеклом».
2. Обучающий курс «Структура данных «Очередь»».
3. Обучающий курс «Динамические деревья».
4. Обучающий курс «Хеширование».
5. Обучающий курс «Задачи на стеках»..
6. Обучающий курс «Линейные списки».

Рекомендации по подготовке к зачету:

При ответе на каждый вопрос зачета студент должен продемонстрировать знание определения указанного понятия, связанных с ним особенностей реализации и применения, умение реализовать указанный алгоритм / структуру

данных, а также навыки иллюстрации теоретических принципов на предложенных простых примерах.

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

1. Компьютерные классы ДВФУ (кампус на о. Русском, Аякс 10)
2. Системное и прикладное обеспечение ПЭВМ.
3. Мультимедийные технические средства.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

**УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ
САМОСТОЯТЕЛЬНОЙ РАБОТЫ**

Основы информатики и программирования
Направление подготовки **09.03.03 Прикладная информатика**
профиль «Прикладная информатика в компьютерном дизайне»
Форма подготовки очная

**Владивосток
2016**

План-график выполнения самостоятельной работы по дисциплине

№ п/п, название	Дата/сроки выполнения	Вид СРС	Примерные нормы времени на выполнение	Форма контроля
1. Стек	Вторая неделя 3 семестра	ИДЗ	1 неделя	Назначение в системе Bb.dvfu
2. Очередь	Четвёртая неделя 3 семестра	ИДЗ	1 неделя	Назначение в системе Bb.dvfu
3. Сортировка	Восьмая неделя 3 семестра	ИДЗ	3 недели	Назначение в системе Bb dvfu
4. Поиск	Одиннадцатая неделя 3 семестра	ИДЗ	3 неделя	Коллоквиум
5. Выражения	Тринадцатая неделя 3 семестра	ИДЗ	1 неделя	Назначение в системе Bb dvfu
6. Калькулятор	Вторая неделя 4 семестра	ИДЗ	2 недели	Назначение в системе Bb.dvfu
7. Движение объектов	Четвертая неделя 4 семестра	ИДЗ	1 неделя	Тестирование
8. Проект программного продукта	Десятая неделя 4 семестра	ИДЗ	3 недели	Письменный отчет
9. Реализация программного	Пятнадцатая неделя	ИДЗ	3 недели	Тестирование

продукта в оконной среде	4 семестра			
10. Презентация программного продукта	Зачётная неделя 4 семестра	ИДЗ	1 час	Собеседование

Материалы для самостоятельной работы студентов подготовлены в виде индивидуальных домашних заданий по каждой теме (образцы типовых ИДЗ представлены в разделе «Материалы для самостоятельной работы студентов»). Работа должна быть отправлена преподавателю на проверку в системе Bb dvfu по соответствующему «Назначению». Оформление в формате PDF. Критерии оценки: студент получает максимальный балл, если работа выполнена без ошибок и оформлена в соответствии с требованиями преподавателя.

ТЕМЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

- Основные концепции динамического распределения памяти
- Методы сортировки и поиска информации.
- Аспекты теории алгоритмизации.
- Изучение статей по дисциплине.
- Использование в современных информационных технологиях алгоритмов сортировки и поиска.

ТЕМЫ ДЛЯ САМОСТОЯТЕЛЬНОЙ ПОДГОТОВКИ К ЗАЧЕТУ

1. Стек. Операции над стеком.
2. Способы реализации стека.
3. Очередь. Операции над очередью.
4. Способы реализации очереди.
5. Дек. Операции над деком.
6. Способы реализации дека.

7. Примеры приложений стека.
8. Примеры приложений очереди.
9. Примеры приложений дека.
10. Дерево. Операции над деревом.
11. Способы реализации дерева.
12. Примеры приложений дерева.
13. Бинарное дерево. Способы реализации.
14. Графы. Способы представления графов.
15. Примеры приложений дерева.
16. Примеры приложений графов.
17. Сортировка линейная.
18. Блочная сортировка.
19. Сортировка перемешиванием.
20. Пирамидальная сортировка.
21. Гномья сортировка.
22. Сортировка двоичным деревом.
23. Сортировка выбором, вставками.
24. «Чет-нечет» сортировка.
25. Сортировка слиянием.
26. Сортировка Шейла.
27. Бинарный поиск.
28. Линейный поиск с барьером.
29. Поиск в ширину.
30. Поиск линейная.
31. Поиск по групповому признаку.
32. Алгоритм «дихотомия».
33. $\alpha - \beta$ отсечение.
34. Метод трихотомии.
35. Индексно-последовательный поиск.

36. Алгоритм Бойлера-Мура
37. Алгоритм Дейкстры.
38. Алгоритм поиска A*.
39. Интерполяционный поиск.
40. Алгоритм поиска A*.
41. Алгоритм «золотого сечения».
42. Алгоритм Бога.
43. Радужные таблицы.
44. Решето Эратосфена.
45. Алгоритм Кнута-Морриса-Пратта.
46. Алгоритм Раббина-Карпа.

КОНСПЕКТЫ ЗАНЯТИЙ

ВВЕДЕНИЕ В ДИСЦИПЛИНУ

Несмотря на то, что с понятием информации мы сталкиваемся ежедневно, строгого и общепризнанного ее определения до сих пор не существует, практически каждая научная дисциплина дает определение по-своему, выделяя в качестве основных компонентов те, которые наилучшим образом соответствуют ее предмету и задачам. При этом типична ситуация, когда понятие об информации, введенное в рамках одной научной дисциплины, может опровергаться конкретными примерами и фактами, полученными в рамках другой науки. Например, представление об информации как о совокупности данных, повышающих уровень знаний об объективной реальности окружающего мира характерное для естественных наук, может быть опровергнуто в рамках социальных наук. Нередки также случаи, когда исходные компоненты, составляющие понятия информации, подменяют свойствами информационных объектов.

Существование множества определений информации обусловлено сложностью, специфичностью и многообразием подходов к толкованию сущности этого понятия. Одна концепция (концепция К.Шеннона), отражает количественно-информационный подход, другая концепция рассматривает информацию как свойство (атрибут) материи, третья основана на логико-семантическом подходе, при котором информация трактуется как специфическое знание.

Таким образом, данный курс в первую очередь играет важную роль в процессе формирования научного мировоззрения учащихся. Значительная часть лекционного материала имеет обзорный характер.

Общепризнанным фактом является также большая роль информатики в развитии мышления. При этом законы самого мышления являются объектом

исследования в рамках данного предмета. Форма проведения занятий курса – коллективный всесторонний анализ предложенной научной задачи в интерактивном режиме, синтез и формализация выводов.

Изучаемая дисциплина формирует основные компетенции специалиста в области информатики, информационных технологий и программирования.

Целью изучения дисциплины «Основы информатики и программирования» является развитие логического и алгоритмического мышления. Привить навыки систематического исследования социальных, технических, экономических и других проблем науки и производства, умение мыслить научными категориями в области науки, техники, экономики и социальной сферы. Студент должен овладеть основными программистскими навыками, необходимыми для решения задач, ознакомиться с современными высокоуровневыми языками программирования; изучить основы алгоритмизации и использовать эти знания при знакомстве с методами программирования. Применение полученных знаний при изучении способов и средств обработки информации.

При подготовке к практическим занятиям следует пользоваться электронными учебными материалами, предложенными студентам электронными ресурсами и рекомендуемой литературой.

По результатам выполненных самостоятельно каждым студентом работ и активности студента на занятиях выставляется итоговая оценка.

Полученные навыки по курсу «Основы информатики и программирования» в дальнейшем будут использоваться при изучении таких дисциплин как: проектирование ИС, технология программирования, операционные системы экономика и управление производством.

Основные понятия

Носитель информации — материальный объект, в некоторых свойствах которого отражаются в течение заметного для наблюдателя времени отдельные свойства иных (не обязательно материальных) объектов.

Машиночитаемый носитель информации (память) — носитель информации, для которого операции чтения/ записи выполняются с помощью автоматических устройств.

Сигнал — состояние носителя информации, имеющее интерпретацию.

Дискретизация — процесс разделения информации на сигналы.

Интерпретация — смысловое значение сигнала.

Машинный код — интерпретация сигнала, выраженная машинными сигналами.

Канал связи — (англ. *channel, data line*) система технических средств и среда распространения сигналов для односторонней передачи данных от источника к получателю.

Информационный процесс — процесс получения, создания, сбора, обработки, накопления, хранения, поиска, распространения и использования информации.

Информационный объем — длина двоичного сообщения.

Структура данных — способ размещения данных в памяти..

Оператор — команда из набора команд языка высокого уровня.

Данные — сообщения, сохраненные на машиночитаемом носителе.

Приложение — Прикладная программа, предназначенная для выполнения определенных пользовательских задач и рассчитанная на непосредственное взаимодействие с пользователем.

Байт — наименьшая адресуемая единица памяти компьютера, равная 8 битам.

Синтаксис — структура оператора.

Семантика — интерпретация оператора.

Подпрограмма — алгоритм, записанный на машинном языке.

Файл — простой набор записей, содержащих логически связанные данные.

Система счисления — система формирования числа из цифр.

Дополнительные теоретические материалы

МЕТОДОЛОГИЯ РАЗРАБОТКИ

ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Среда разработки Microsoft Visual Studio (Visual Studio.NET) предоставляет мощные и удобные средства написания, корректировки, компиляции, отладки и запуска приложений, использующих .NET-совместимые языки. Корпорация Microsoft включила в платформу средства разработки для четырех языков: Visual C#, Visual J#, Visual Basic и Visual C++. Платформа .NET является открытой средой. Это означает, что компиляторы для нее могут поставляться и сторонними разработчиками. К настоящему времени разработаны десятки компиляторов для .NET, например, Ada, Perl, Delphi, Lisp и другие. Все .NET-совместимые языки должны отвечать требованиям CLS (Common Language Specification – общезыко́вая спецификация), в которой описывается набор общих для всех них характеристик. Это позволяет использовать для разработки приложения несколько языков программирования и вести полноценную межъязыковую отладку. Все программы, созданные в Visual Studio.NET, независимо от языка программирования на котором они написаны, используют одни и те же базовые классы библиотеки .NET.

Приложение в процессе разработки называется проектом. Проект включает в себя все необходимое для создания приложения: файлы, папки, ссылки и прочие ресурсы. Среда Visual Studio.NET позволяет создавать проекты различных типов, таких как:

1) Windows-приложение использует элементы интерфейс ОС Windows, включая формы, кнопки, флажки и т.п.

2) Консольное приложение.

3) Библиотеки классов. Они объединяют в себе классы, которые предназначены для использования в других приложениях.

4) Web-приложение. Это приложение, доступ к которому осуществляется через браузер, и которое по запросу формирует web-страницу и отправляет ее клиенту по сети.

5) Web-сервис. Это компонент, методы которого могут вызываться через Интернет.

Несколько проектов можно объединить в решение (solution), что облегчает их совместную разработку. Скомпилированная программа будет выполняться ОС Windows, только если будет установлена исполняющая среда .NET (dot-netfx.exe). Этот пакет поставляется совместно с Visual Studio и его можно установить в любой версии ОС Windows, начиная с Windows'98. Исполняемая среда контролирует исполнение программ (управляемый код), сглаживая некоторые ошибки.

ПОЛИМОРФИЗМ

Класс может являться наследником одного или нескольких базовых классов. Это означает, что данный класс содержит как часть описание всех базовых классов, с их структурой данных и методами. В процессе конструирования класса-наследника сначала вызываются конструкторы базовых классов, и только затем его собственный конструктор. Деструкторы вызываются в обратном порядке. Методы класса могут быть объявлены как виртуальные (динамические). Метод, объявленный как виртуальный, вызывается для объекта не по типу указателя, а по фактическому типу содержащегося по данному указателю объекта.

Наследование, вместе с инкапсуляцией и полиморфизмом, является одной из трех основных характеристик (или базовых понятий) объектно-ориентированного программирования. Наследование позволяет создавать новые классы, которые повторно используют, расширяют и изменяют

поведение, определенное в других классах. Класс, члены которого наследуются, называется базовым классом, а класс, который наследует эти члены, называется производным классом. Производный класс может иметь только один непосредственный базовый класс. Однако наследование является транзитивным. Если ClassC является производным от ClassB, и ClassB является производным от ClassA, ClassC наследует члены, объявленные в ClassB и ClassA.

Концептуально, производный класс является специализацией базового класса. Например, при наличии базового класса Animal, возможно наличие одного производного класса, который называется Mammal, и еще одного производного класса, который называется Reptile. Mammal является Animal, а Reptile является Animal, но каждый производный класс представляет разные специализации базового класса.

При определении класса для наследования от другого класса, производный класс явно получает все члены базового класса, за исключением его конструкторов и деструкторов. Производный класс может таким образом повторно использовать код в базовом классе без необходимости в его повторной реализации. В производном классе можно добавить больше членов. Таким образом, производный класс расширяет функциональность базового класса.

Ниже иллюстрируется класс WorkItem, представляющий рабочий элемент в бизнес-процессе. Подобно всем классам, он является производным от System.Object и наследует все его методы. В WorkItem имеется пять собственных членов. Сюда входит конструктор, поскольку конструкторы не наследуются. Класс ChangeRequest наследуется от WorkItem и представляет конкретный вид рабочего элемента. ChangeRequest добавляет еще два члена к членам, унаследованным от WorkItem и Object. Он должен добавить собственный конструктор, и он также добавляет originalItemID.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего
образования
«Дальневосточный федеральный университет»
(ДВФУ)

НАЗВАНИЕ ШКОЛЫ (ФИЛИАЛА)

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

Основы информатики и программирования
Направление подготовки **09.03.03 Прикладная информатика**
профиль «Прикладная информатика в компьютерном дизайне»
Форма подготовки очная

Владивосток
2016

Паспорт ФГОС

Код и формулировка компетенции	Этапы формирования компетенции	
ОК-3 Способность проявлять инициативу и принимать ответственные решения, осознавая ответственность за результаты своей профессиональной деятельности	Знает	Понятие и назначение формализации. Математические методы в формализации задачи.
	Умеет	Применять математические (численные) методы при решении задач.
	Владеет	Навыками научно-исследовательской деятельности: системному подходу к исследуемой проблеме по поставленному заданию. Формализации постановки задачи.
ПК-2 Способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение	Знает	Основные этапы постановки задачи. Набор инструментальных средств, достаточный для решения задач средней сложности.
	Умеет	Формировать неформальную и формальную постановку задачи. Обосновывать необходимость решения задачи.
	Владеет	Навыками разработки и адаптации программного обеспечения средней сложности.
ПК-13 Способностью проводить тестирование компонентов программного обеспечения	Знает	Основные этапы коллективной разработки программного продукта, технологию тестирования программного продукта.
	Умеет	Обеспечивать формирование наборов тестов для проверки программного продукта.
	Владеет	Навыками работы в интегрированной среде; методами алгоритмизации и программирования; навыками тестирования и отладки приложений.
ПК-15 Способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач	Знает	Основы теории баз данных.
	Умеет	Использовать основные законы и требования к разработке и ведению баз данных в профессиональной деятельности.
	Владеет	Навыками проектирования и реализации баз данных при решении прикладных задач.

Шкала оценивания уровня сформированности компетенций

ОК-3 Способность проявлять инициативу и принимать ответственные решения, осознавая ответственность за результаты своей профессиональной деятельности

Планируемые результаты обучения* (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения**				
	1	2	3	4	5
Знает: Основные понятия и законы информатики, виды информации, основные информационные процессы, современный взгляд на роль информации.	Отсутствие знания основных понятий и законов информатики, видов информации, основных информационных процессов, современного взгляда на роль информации.	Фрагментарное знание основных понятий и законов информатики, виды информации, основные информационные процессы, современный взгляд на роль информации,	Неполное знание основных понятий и законов информатики, видов информации, основных информационных процессов, современного взгляда на роль информации,	В целом сформированное знание современного представления об основных понятиях и законах информатики, видах информации, основных информационных процессов, современного взгляда на роль информации.	Сформированное систематическое знание современного представления об основных понятиях и законах информатики, видах информации, основных информационных процессов, современного взгляда на роль информации.
Умеет: Оценивать параметры информационных сообщений, рассчитывать информационный объем, параметры кодирования информации, обобщать свойства информации, анализировать способы ее кодирования.	Отсутствует умение оценивать параметры информационных сообщений, рассчитывать информационный объем, параметры кодирования информации, обобщать свойства информации, анализировать	Фрагментарное умение оценивать параметры информационных сообщений, рассчитывать информационный объем, параметры кодирования информации, обобщать свойства информации, анализировать	Неполное умение оценки параметров информационных сообщений, рассчитывать информационный объем, параметры кодирования информации, обобщать свойства информации, анализировать способы ее кодирования.	В целом сформированное умение оценки параметров информационных сообщений, рассчитывать информационный объем, параметры кодирования информации, обобщать свойства информации, анализировать	Сформированное систематическое умение оценки параметров информационных сообщений, рассчитывать информационный объем, параметры кодирования информации, обобщать свойства

	ь способы ее кодирования.	ь способы ее кодирования.		анализировать способы ее кодирования.	информации, анализировать способы ее кодирования.
Владеет: Современными математическими средствами оценки параметров информатики и ее кодирования.	Отсутствие владения современным и математическими средствами оценки параметров информатики и ее кодирования.	Фрагментарное владение современным и математическими средствами оценки параметров информатики и ее кодирования.	Неполное владение современными математическими средствами оценки параметров информатики и ее кодирования.	В целом сформировавшаяся владение современным и математическими средствами оценки параметров информатики и ее кодирования.	Сформировавшаяся систематическое владение современным и математическими средствами оценки параметров информатики и ее кодирования.
Шкала оценивания (соотношение с традиционным и формами аттестации)	0–8 неудовлетворительно	9–12 неудовлетворительно	13–15 удовлетворительно	16–18 хорошо	19–20 отлично

ПК-2 Способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение

Планируемые результаты обучения* (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения**				
	1	2	3	4	5
Знает: Основные этапы коллективной разработки программного продукта, требования к участникам коллективной разработки программного продукта.	Отсутствие знания основных этапов коллективной разработки программного продукта, требования к участникам коллективной разработки программного продукта.	Фрагментарное знание основных этапов коллективной разработки программного продукта, требования к участникам коллективной разработки программного продукта.	Неполное знание основных этапов коллективной разработки программного продукта, требования к участникам коллективной разработки программного продукта.	В целом сформировавшееся знание современного представления об основных этапах коллективной разработки программного продукта, требования к участникам коллективной разработки	Сформировавшееся систематическое знание современного представления об основных этапах коллективной разработки программного продукта, требования к участникам коллективной

				программного продукта.	разработки программного продукта.
Умеет: Обеспечивать поддержку коллективной разработки и модификации программного продукта.	Отсутствие умения обеспечивать поддержку коллективной разработки и модификации программного продукта.	Фрагментарное умение обеспечивать поддержку коллективной разработки и модификации программного продукта.	Неполное умение обеспечивать поддержку коллективной разработки и модификации программного продукта.	В целом сформированное умение обеспечивать поддержку коллективной разработки и модификации программного продукта.	Сформированное умение обеспечивать поддержку коллективной разработки и модификации программного продукта способами ее кодирования.
Владеет: Начальными навыками защиты разрабатываемого программного продукта от непрофессиональной, некорректной модификации.	Отсутствие владения начальными навыками защиты разрабатываемого программного продукта от непрофессиональной, некорректной модификации.	Фрагментарное владение начальными навыками защиты разрабатываемого программного продукта от непрофессиональной, некорректной модификации.	Неполное владение начальными навыками защиты разрабатываемого программного продукта от непрофессиональной, некорректной модификации.	В целом сформированное владение начальными навыками защиты разрабатываемого программного продукта от непрофессиональной, некорректной модификации.	Сформированное владение начальными навыками защиты разрабатываемого программного продукта от непрофессиональной, некорректной модификации.
Шкала оценивания (соотношение с традиционными и формами аттестации)	0–8 неудовлетворительно	9–12 неудовлетворительно	13–15 удовлетворительно	16–18 хорошо	19–20 отлично

ПК-13 Способностью проводить тестирование компонентов программного обеспечения

Планируемые результаты обучения* (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения**				
	1	2	3	4	5
Знает: Средства работы с	Отсутствие знания средств	Фрагментарное знание средств	Неполное знание средств работы с	В целом сформированное знание	Сформированное систематическое

информацией в глобальных компьютерных сетях, основные принципы доступа к библиотечным ресурсам.	работы с информацией в глобальных компьютерных сетях, основных принципов доступа к библиотечным ресурсам.	работы с информацией в глобальных компьютерных сетях, основных принципов доступа к библиотечным ресурсам.	информацией в глобальных компьютерных сетях, основных принципов доступа к библиотечным ресурсам.	средств работы с информацией в глобальных компьютерных сетях, основных принципов доступа к библиотечным ресурсам.	кое знание средств работы с информацией в глобальных компьютерных сетях, основных принципов доступа к библиотечным ресурсам.
Умеет: Самостоятельно осваивать отдельные темы дисциплины под минимальным руководством преподавателя, с использованием методических указаний. Самостоятельно оценивать надежность и работоспособность программного продукта средней сложности.	Отсутствие умения самостоятельно осваивать отдельные темы дисциплины под минимальным руководством преподавателя, с использованием методических указаний. Самостоятельно оценивать надежность и работоспособность программного продукта средней сложности.	Фрагментарное умение самостоятельно осваивать отдельные темы дисциплины под минимальным руководством преподавателя, с использованием методических указаний. Самостоятельно оценивать надежность и работоспособность программного продукта средней сложности.	Неполное умение самостоятельно осваивать отдельные темы дисциплины под минимальным руководством преподавателя, с использованием методических указаний. Самостоятельно оценивать надежность и работоспособность программного продукта средней сложности.	В целом сформированное умение самостоятельно осваивать отдельные темы дисциплины под минимальным руководством преподавателя, с использованием методических указаний. Самостоятельно оценивать надежность и работоспособность программного продукта средней сложности.	Сформированное умение систематическое умение самостоятельно осваивать отдельные темы дисциплины под минимальным руководством преподавателя, с использованием методических указаний. Самостоятельно оценивать надежность и работоспособность программного продукта средней сложности.
Владеет: Навыками работы в интегрированной среде; методами алгоритмизации и программирования; навыками тестирования и	Отсутствие владения навыками работы в интегрированной среде; методами алгоритмизации и программирования; навыками	Фрагментарное владение навыками работы в интегрированной среде; методами алгоритмизации и программирования; навыками	Неполное владение навыками работы в интегрированной среде; методами алгоритмизации и программирования; навыками	В целом сформированное владение навыками работы в интегрированной среде; методами алгоритмизации и программирования	Сформированное систематическое владение навыками работы в интегрированной среде; методами алгоритмизации и программирования

отладки приложений.	тестирования и отладки приложений.	тестирования и отладки приложений.	тестирования и отладки приложений.	ания; навыками тестирования и отладки приложений.	ания; навыками тестирования и отладки приложений.
Шкала оценивания (соотношение с традиционным и формами аттестации)	0–8 неудовлетворительно	9–12 неудовлетворительно	13–15 удовлетворительно	16–18 хорошо	19–20 отлично

ПК-15 Способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач

Планируемые результаты обучения* (показатели достижения заданного уровня освоения компетенций)	Критерии оценивания результатов обучения**				
	1	2	3	4	5
Знает: Социальную значимость информации в развитии современного информационного общества, ее долю в профессиональной деятельности людей, тенденции изменения ее роли в обществе. Угрозы для общества, возникающие в процессе усиления роли информации в его развитии. Правовую основу использования	Отсутствие знания социальной значимости информации в развитии современного информационного общества, ее долю в профессиональной деятельности людей, тенденции изменения ее роли в обществе. Угрозы для общества, возникающие в процессе усиления роли информации в его развитии.	Фрагментарное знание социальной значимости информации в развитии современного информационного общества, ее долю в профессиональной деятельности людей, тенденции изменения ее роли в обществе. Угрозы для общества, возникающие в процессе усиления роли информации в его развитии.	Неполное знание социальной значимости информации в развитии современного информационного общества, ее долю в профессиональной деятельности людей, тенденции изменения ее роли в обществе. Угрозы для общества, возникающие в процессе усиления роли информации в его развитии. Правовую основу	В целом сформированное знание социальной значимости информации в развитии современного информационного общества, ее долю в профессиональной деятельности людей, тенденции изменения ее роли в обществе. Угрозы для общества, возникающие в процессе усиления роли информации в	Сформированное систематическое знание социальной значимости информации в развитии современного информационного общества, ее долю в профессиональной деятельности людей, тенденции изменения ее роли в обществе. Угрозы для общества, возникающие в процессе усиления роли

информации.	Правовую основу использования информации.	Правовую основу использования информации.	использования информации.	его развитии. Правовую основу использования информации.	информации в его развитии. Правовую основу использования информации.
Умеет: Использовать основные способы защиты информации.	Отсутствие умения использования основных способов защиты информации.	Фрагментарное умение использования основных способов защиты информации.	Неполное использования основных способов защиты информации.	В целом сформированное умение использования основных способов защиты информации.	Сформированное систематическое умение использования основных способов защиты информации.
Владеет: Техническими и программными средствами защиты информации.	Отсутствие владения техническими и программными средствами защиты информации.	Фрагментарное владение техническими и программными средствами защиты информации.	Неполное владение техническими и программными средствами защиты информации.	В целом сформированное владение техническими и программными средствами защиты информации.	техническими и программными средствами защиты информации.
Шкала оценивания (соотношение с традиционными и формами аттестации)	0–8 неудовлетворительно	9–12 неудовлетворительно	13–15 удовлетворительно	16–18 хорошо	техническими и программными средствами защиты информации.

Оценочные средства для промежуточной аттестации

1. Эффективное применение инкапсуляции в объектно-ориентированном программировании.
3. Открытый интерфейс и эффективное сокрытие реализации.
4. Использование концепции наследования при объектно-ориентированном программировании.
5. Применение простого наследования.
6. Использование абстрактных классов при наследовании.
7. Адаптация объектно-ориентированных программ к изменяющимся требованиям средствами полиморфизма.

8. Применение полиморфизма включения.
9. Переопределение – важный тип полиморфизма.
10. Перегрузка – частный случай полиморфизма.
11. Необходимое условие эффективного полиморфизма – эффективное применение инкапсуляции и наследования.
12. Построение UML-диаграмм классов программных продуктов, разрабатываемых в домашних и аудиторных работах.
13. Анализ прецедентов – случаев взаимодействия пользователя с системой – при выполнении объектно-ориентированного анализа с целью уяснения смысла задач, разрабатываемых в домашних и аудиторных работах.
14. Построение концептуальной модели – выявление объектов предметной области, необходимых для адекватного описания системы – при выполнении объектно-ориентированного анализа с целью уяснения смысла задач, разрабатываемых в домашних и аудиторных работах.
15. Построение объектной модели – установление взаимосвязей и структуры объектов – при выполнении объектно-ориентированного проектирования с целью приближения к искомой конструкции систем, разрабатываемых в домашних и аудиторных работах.
16. Практическое применение шаблонов при выполнении объектно-ориентированного программирования.
17. Программирование пользовательского интерфейса на основе объектно-ориентированного подхода.

Оценочные средства для текущей аттестации

- Значение информации в развитии современного информационного общества.
- Тенденции и проблемы развития современного информационного общества.
- Основные понятия информатики.
- Виды и формы информации. Информационные процессы.
- Технические и программные средства реализации информационных процессов.
- Принципы кодирования информации.

- Машиночитаемые носители информации. Правила кодирования.
- Определение информационного объема.
- Опасности и угрозы информационного общества.
- Основы защиты информации
- Структура программного обеспечения.
- Типы и структуры данных.
- Потоки управления в программах.
- Базовые алгоритмы обработки информации. Оценка сложности и эффективности алгоритмов.
- Структура программ.
- Стил программирования.
- Этапы создания программ.
- Статическое распределение памяти.
- Динамическое распределение памяти.
- Динамические структуры данных: стек, очередь, список, дерево.
- Языки программирования высокого уровня.
- Базовые типы данных.
- Составные типы данных: массив, строка, запись.
- Целочисленная арифметика.
- Организация и средства человеко-машинного интерфейса.
- Основные алгоритмические конструкции.
- Базовые алгоритмы.
- Подпрограммы: процедуры, функции.
- Двумерные массивы.
- Алгоритмы сортировки.
- Организация работы с файлами
- Метод рекурсии.
- Алгоритмы поиска.
- Метод перебора.
- Технология тестирования.
- Синтаксис и семантика языковых конструкций.



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«Дальневосточный федеральный университет»
(ДВФУ)

ШКОЛА ЕСТЕСТВЕННЫХ НАУК

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по дисциплине «Основы информатики и программирования»
Направление подготовки 09.03.03 Прикладная информатика
профиль/ специализация/ магистерская программа «Прикладная
информатика в компьютерном дизайне»
Форма подготовки очная

Владивосток
2016

СОРТИРОВКА С ПОМОЩЬЮ БИНАРНОГО ДЕРЕВА

Использованы материалы: Методы сортировки и поиска.
С.Д. Кузнецов, ИСП РАН, Центр Информационных Технологий

Опишем один из простых методов сортировки, в котором явно строится бинарное дерево «сравнения ключей». Построение дерева начинается с листьев, которые содержат все элементы массива. Из каждой соседней пары выбирается наименьший элемент, и эти элементы образуют следующий (более высокий) уровень дерева. Из каждой соседней пары выбирается наименьший элемент и т.д., пока не будет построен корень, содержащий наименьший элемент массива. Двоичное дерево сравнения для массива, используемого в наших примерах, показано на рисунке 1. В момент завершения построения дерева мы уже имеем минимальное значение массива. Чтобы получить следующий наименьший по величине элемент, перейдем от корня по пути, ведущему к листу с наименьшим значением. В этой листовой вершине значение заменяется на «фиктивный ключ» с "невозможно большим" значением, а во все промежуточные узлы, занимавшиеся наименьшим элементом, заносится наименьшее значение из узлов - непосредственных потомков (рис. 2). Процесс продолжается до тех пор, пока все узлы дерева не будут заполнены «фиктивными ключами» (рис. 3 – 8).

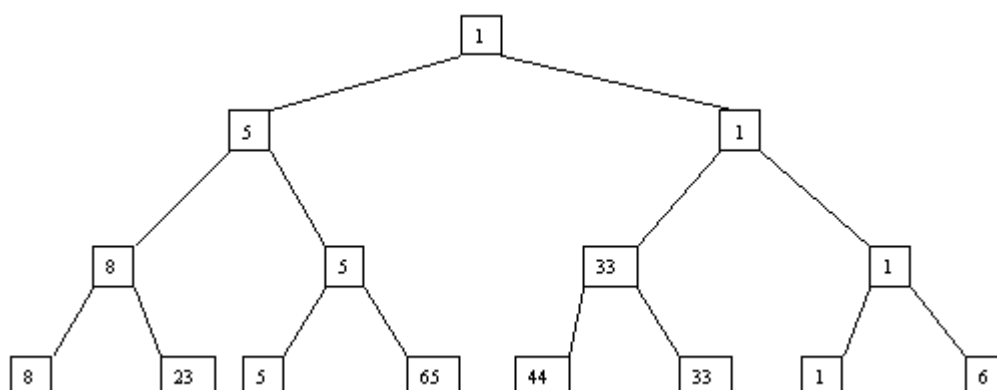


Рис. 1. Построенное дерево

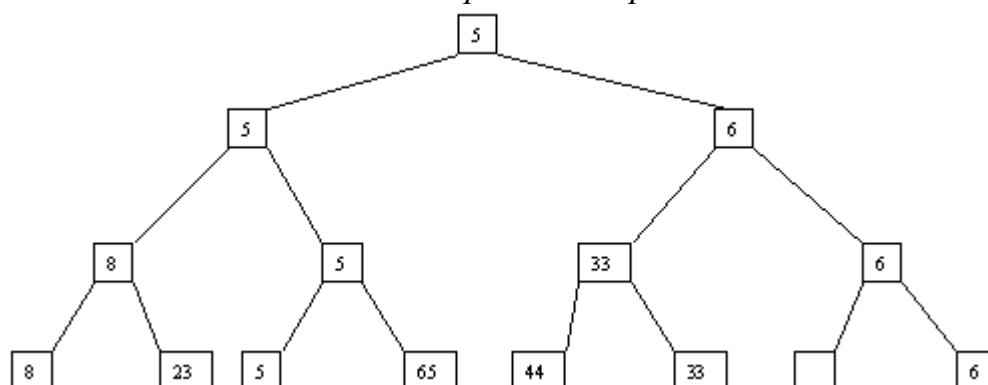


Рис. 2 Шаг второй

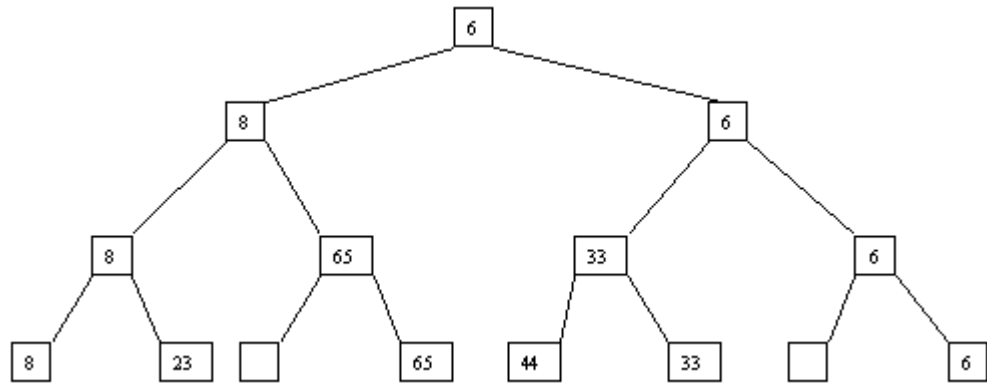


Рис. 3. Шаг третий

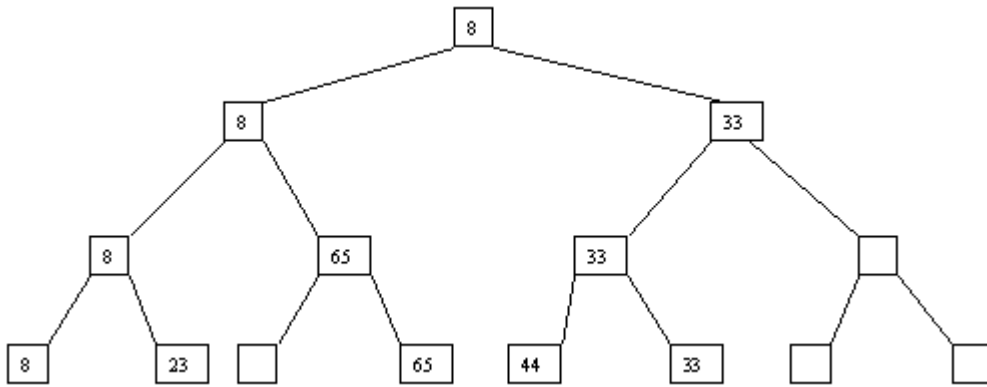


Рис. 4. Шаг четвертый

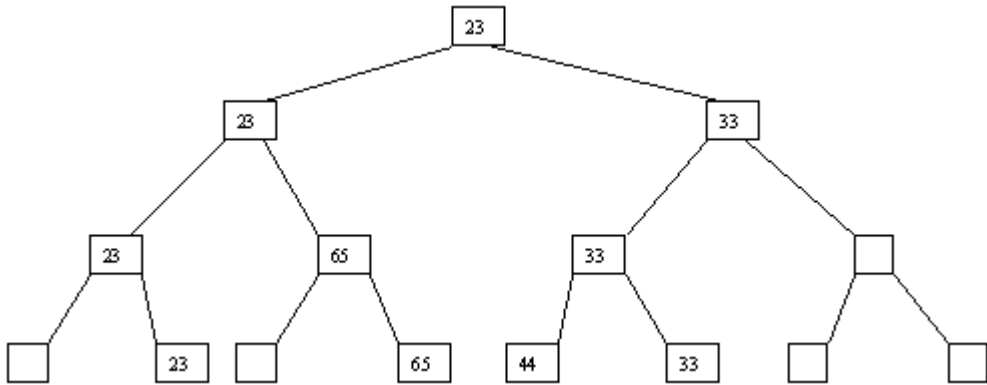


Рис. 5. Шаг пятый

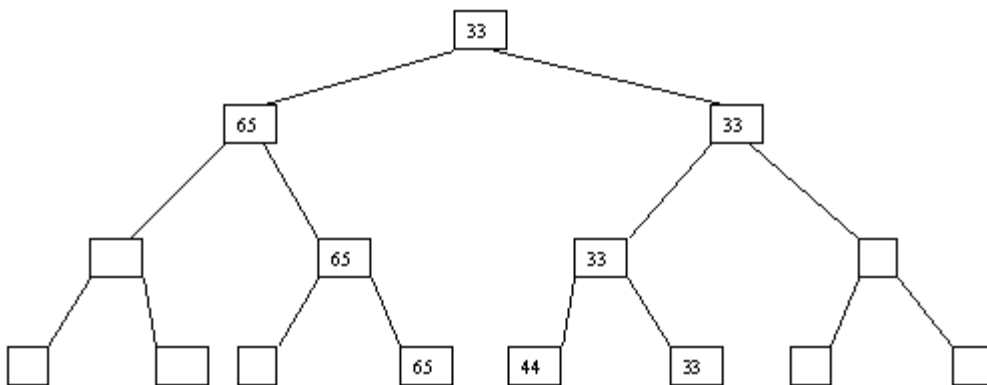


Рис. 2.6. Шестой шаг

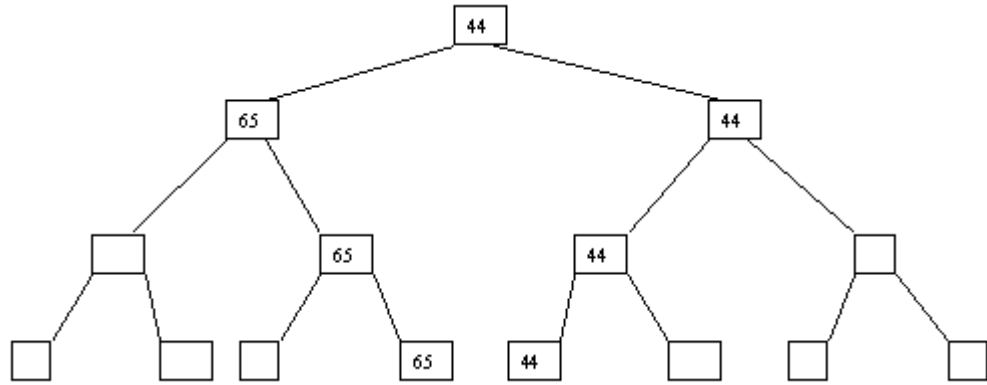


Рис. 7. Шаг седьмой

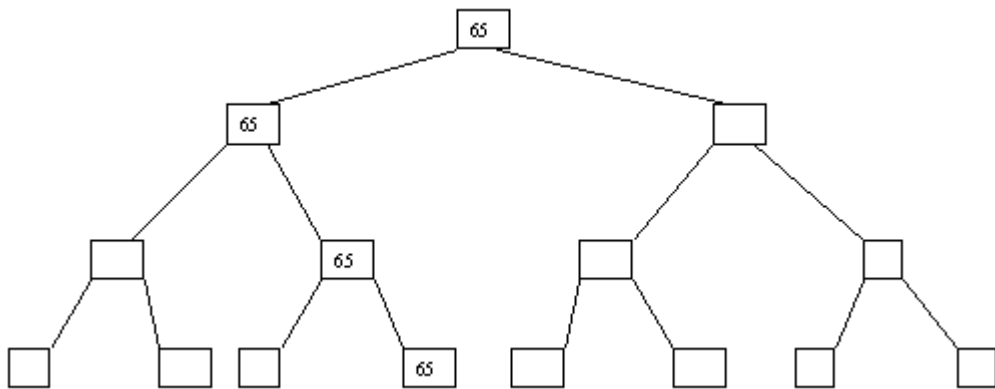


Рис. 2.8. Восьмой шаг

На каждом из n шагов, требуемых для сортировки массива, требуется $\log_2 n$ сравнений. Соответственно, всего потребуется $n \cdot \log_2 n$ сравнений. Для представления дерева необходимо выделение $2n-1$ дополнительных единиц памяти.

Существует более совершенный алгоритм, называемый Пирамидальной сортировкой. Его идея состоит в том, что вместо полного дерева сравнения исходный массив $a[1], a[2], \dots, a[n]$ преобразуется в *пирамиду*, в которой, что для каждого элемента $a[i]$ выполняются следующие условия $a[i] \leq a[2i]$ и $a[i] \leq a[2i+1]$. После чего пирамида используется для сортировки.

Метод построения пирамиды более наглядно выглядит при представлении массива через дерево, см. рисунок 9. Здесь массив представляется в форме бинарного дерева, в корне которого находится элемент массива $a[1]$. На втором уровне стоят элементы $a[2]$ и $a[3]$. На третьем, соответственно: $a[4]$, $a[5]$, $a[6]$, $a[7]$ и т.д. Очевидно, дерево будет сбалансированным для массива с нечетным количеством элементов, а для массива с четным количеством элементов – "почти" сбалансированным, в котором элемент $a[n]$ будет самым левым, единственным листом последнего уровня.

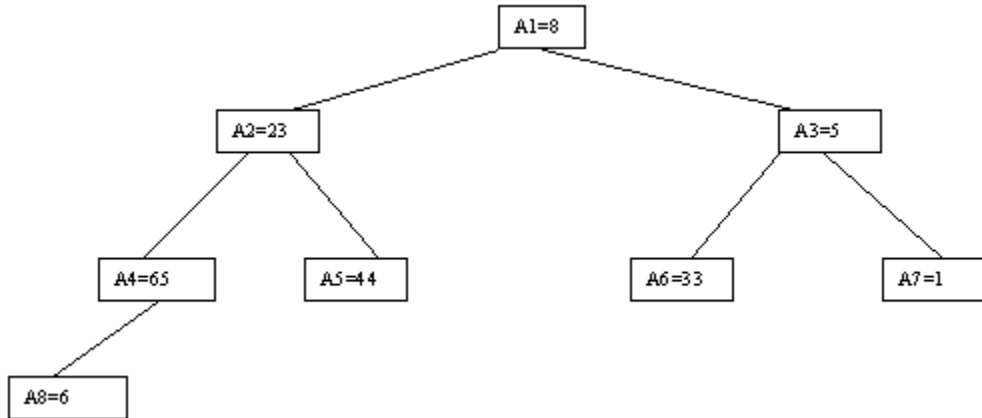


Рис. 9.

При построении пирамиды для массивов с четным числом элементов мы будем рассматривать элементы: $a[\lfloor n/2 \rfloor]$, $a[\lfloor n/2 \rfloor - 1]$, ..., $a[1]$, а для массивов с нечетным числом элементов: $a[\lfloor (n-1)/2 \rfloor]$, $a[\lfloor (n-1)/2 \rfloor - 1]$, ..., $a[1]$. Если i – наибольший индекс из числа индексов элементов, для которых существенны ограничения пирамиды. Тогда берется элемент $a[i]$ в построенном дереве и для него выполняется следующая процедура: выбирается ветвь дерева, соответствующая $\min(a[2i], a[2i+1])$, и значение $a[i]$ меняется местами со значением соответствующего элемента. Пока этот элемент не является листом дерева, для него повторяется аналогичная процедура. Такие действия выполняются последовательно для $a[i]$, $a[i-1]$, ..., $a[1]$. В результате мы получим представление пирамиды в виде дерева для заданного массива (На рисунках 10 – 13 показаны шаги этого процесса).

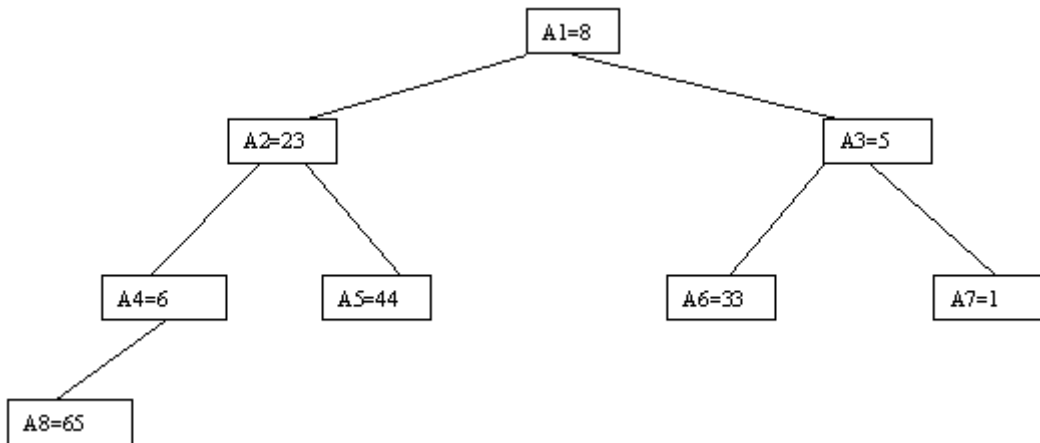


Рис. 10.

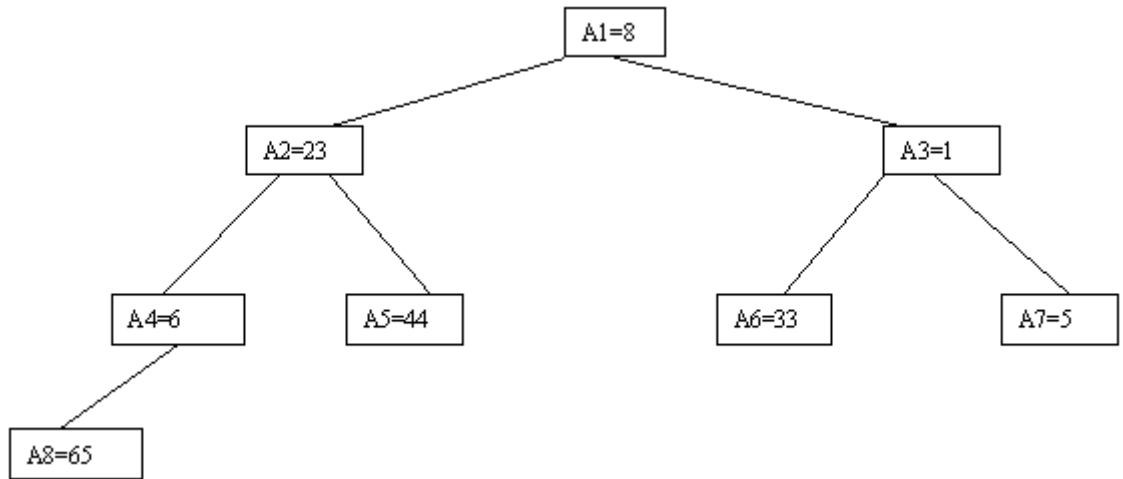


Рис. 11.

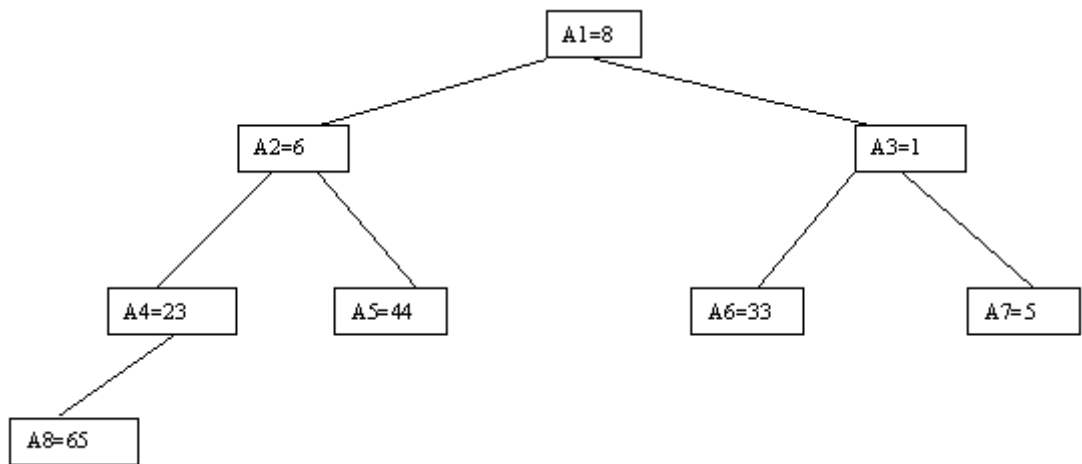


Рис. 12.

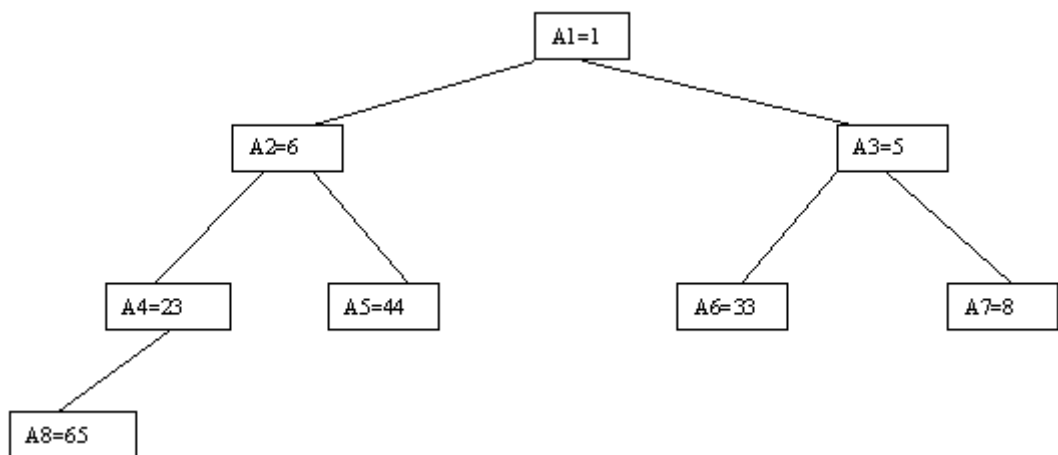


Рис. 13.

Существует также метод Флойда – построение пирамиды без явного построения дерева. Построение пирамиды методом для нашего стандартного массива показано в таблице 1.

Таблица 1. Последовательность построения пирамиды

Начальное состояние массива	8 23 5 65 44 33 1 6
Шаг 1	8 23 5 6 44 33 1 65
Шаг 2	8 23 1 6 44 33 5 65
Шаг 3	8 6 1 23 44 33 5 65
Шаг 4	1 6 8 23 44 33 5 65 1 6 5 23 44 33 8 65

В таблице 2 показано, как производится сортировка с использованием построенной пирамиды. Суть алгоритма заключается в следующем. Пусть i - наибольший индекс массива, для которого существенны условия пирамиды. Тогда начиная с $a[1]$ до $a[i]$ выполняются следующие действия. На каждом шаге выбирается последний элемент пирамиды (в нашем случае первым будет выбран элемент $a[8]$). Его значение меняется со значением $a[1]$, после чего для $a[1]$ выполняется просеивание. При этом на каждом шаге число элементов в пирамиде уменьшается на 1 (после первого шага в качестве элементов пирамиды рассматриваются $a[1], a[2], \dots, a[n-1]$; после второго - $a[1], a[2], \dots, a[n-2]$ и т.д., пока в пирамиде не останется один элемент). Легко видеть по таблице 2, что в результате мы получим массив, упорядоченный в порядке убывания. Можно модифицировать метод построения пирамиды и сортировки, чтобы получить упорядочение в порядке возрастания, если изменить условие пирамиды на $a[i] \geq a[2 \cdot i]$ и $a[1] \geq a[2 \cdot i + 1]$ для всех осмысленных значений индекса i .

Таблица 2. Последовательность сортировки с помощью пирамиды

Исходная пирамида	1 6 5 23 44 33 8 65
Шаг 1	65 6 5 23 44 33 8 1 5 6 65 23 44 33 8 1 5 6 8 23 44 33 65 1
Шаг 2	65 6 8 23 44 33 5 1 6 65 8 23 44 33 5 1 6 23 8 65 44 33 5 1
Шаг 3	33 23 8 65 44 6 5 1 8 23 33 65 44 6 5 1
Шаг 4	44 23 33 65 8 6 5 1 23 44 33 65 8 6 5 1
Шаг 5	65 44 33 23 8 6 5 1 33 44 65 23 8 6 5 1
Шаг 6	65 44 33 23 8 6 5 1 44 65 33 23 8 6 5 1
Шаг 7	65 44 33 23 8 6 5 1

Метод сортировки с использованием пирамиды требует выполнения порядка $n \cdot \log_2 n$ шагов, что делает его наиболее применимым для сортировки больших массивов.