

Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КТПИ ФГБОУ ВО «КемГУ»
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Дата и время: 2024-02-21 00:00:00
Кузбасский государственный технический университет
федерального государственного бюджетного образовательного учреждения
высшего образования
«Кемеровский государственный университет»
Факультет информатики, математики и экономики

УТВЕРЖДАЮ

Декан А.В.Фомина
«10» февраля 2022 г.

Рабочая программа дисциплины

Б1.О.04 Практикум по программированию

Направление

09.03.03 Прикладная информатика

Направленность (профиль) подготовки
«Прикладная информатика в экономике»

Программа бакалавриата

Квалификация выпускника

бакалавр

Форма обучения

очная

год набора 2021

Новокузнецк 2022

Оглавление

1 Цель дисциплины.....	3
1.1 Формируемые компетенции	3
1.2 Индикаторы достижения компетенций	3
1.3 Знания, умения, навыки (ЗУВ) по дисциплине.....	3
2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.....	4
3. Учебно-тематический план и содержание дисциплины.	4
3.1 Учебно-тематический план.....	4
3.2. Содержание занятий по видам учебной работы	5
4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.	6
5 Материально-техническое, программное и учебно-методическое обеспечение дисциплины. ..	7
5.1 Учебная литература	7
5.2 Материально-техническое и программное обеспечение дисциплины.	8
5.3 Современные профессиональные базы данных и информационные справочные системы. ...	11
6 Иные сведения и (или) материалы.	11
6.1. Примерные вопросы и задания / задачи для промежуточной аттестации	11

1 Цель дисциплины.

В результате освоения данной дисциплины у обучающегося должны быть сформированы компетенции основной профессиональной образовательной программы бакалавриата (далее - ОПОП): ОПК–7.

Содержание компетенций как планируемых результатов обучения по дисциплине см. таблицы 1, 2 и 3.

1.1 Формируемые компетенции

Таблица 1 - Формируемые дисциплиной компетенции

Наименование вида компетенции	Наименование категории (группы) компетенций	Код и название компетенции
Общепрофессиональная		ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения

1.2 Индикаторы достижения компетенций

Таблица 2 – Индикаторы достижения компетенций, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции по ОПОП	Дисциплины и практики, формирующие компетенцию ОПОП
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК 7.1 Осуществляет и обосновывает выбор стандартных алгоритмов и программных средств для реализации практических задач ОПК 7.2 Разрабатывает алгоритмы и программы для реализации прикладных практических задач	Б1.О.04 Практикум по программированию Б1.О.07 Программирование Б2.О.01(У) Ознакомительная практика Б2.О.03(П) Эксплуатационная практика Б3.01(Д) Выполнение и защита выпускной квалификационной работы

1.3 Знания, умения, навыки (ЗУВ) по дисциплине

Таблица 3 – Знания, умения, навыки, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции, закрепленные за дисциплиной	<i>Знания, умения, навыки (ЗУВ), формируемые дисциплиной</i>
ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК 7.1 Осуществляет и обосновывает выбор стандартных алгоритмов и программных средств для реализации практических задач ОПК 7.2 Разрабатывает алгоритмы и программы для реализации прикладных практических задач	Знать: <ul style="list-style-type: none">простые и сложные типы данных;понятия и методы алгоритмизации;основы теории алгоритмов и основы теории сложности. Уметь: <ul style="list-style-type: none">разрабатывать алгоритмы для решения прикладных практических задач;разрабатывать программы для реализации прикладных практических задач;обосновывать выбор стандартных алгоритмов для решения практических задач Владеть: <ul style="list-style-type: none">методами алгоритмизации, оценки

Код и название компетенции	Индикаторы достижения компетенции, закрепленные за дисциплиной	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
		сложности алгоритмов; • графическим способом описания алгоритмов; • навыками реализации алгоритмов в программном коде.

2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.

Таблица 4 – Объём и трудоёмкость дисциплины по видам учебных занятий

Общая трудоёмкость и виды учебной работы по дисциплине, проводимые в разных формах	Объём часов по формам обучения		
	ОФО	ОЗФО	ЗФО
1 Общая трудоёмкость дисциплины	144		
2 Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	54		
Аудиторная работа (всего):			
в том числе:			
лекции			
практические занятия, семинары	54		
практикумы			
лабораторные работы			
в интерактивной форме			
в электронной форме			
Внеаудиторная работа (всего):			
в том числе, индивидуальная работа обучающихся с преподавателем			
подготовка курсовой работы /контактная работа ¹			
групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем			
творческая работа (эссе)			
3 Самостоятельная работа обучающихся (всего)	54		
4 Промежуточная аттестация обучающегося – экзамен -3 семестр	36		

3. Учебно-тематический план и содержание дисциплины.

3.1 Учебно-тематический план

Таблица 5 - Учебно-тематический план

№ недели п/п	Разделы и темы дисциплины по занятиям	Общая трудоёмкость (всего час.)	Трудоёмкость занятий (час.)		Формы текущего контроля и промежуточной аттестации
			ОФО		
			Аудиторные занятия	СРС	

¹ Часы, выделенные в УП на курсовое проектирование в контактной форме (3 часа)

			лекц.	практ	ла б.		успеваемости
	Семестр 3	144		54		54	
1-4	1. Типы данных. Простые типы данных 1.1. Числовые типы данных 1.2. Логические типы данных 1.3. Другие типы данных	27	0	10		17	Устный опрос, решение учебных задач, защита отчета о выполнении практической работы
5-8	2. Типы данных. Структурные типы данных 2.1. Записи 2.2. Массивы 2.3. Строки 2.4. Множества 2.5. Стеки 2.6. Очереди 2.7. Списки 2.8. Деревья	27		18		9	Устный опрос, решение учебных задач, защита отчета о выполнении практической работы
9-13	3. Понятие алгоритма. 3.1 Свойства алгоритмов 3.1 Классификация алгоритмов по форме представления, по структуре	27		10		17	Устный опрос, решение учебных задач, защита отчета о выполнении практической работы
14-18	4. Алгоритмы. Понятие сложности алгоритмов 4.1 Определение сложности алгоритма 4.2 Способы определения сложности алгоритмов 4.3 Сложность рекурсивных алгоритмов. Простая рекурсия 4.4 Сложность рекурсивных алгоритмов. Сложная рекурсия 4.5 Функции, используемые для определения сложности алгоритмов	27		16		11	Устный опрос, решение учебных задач, защита отчета о выполнении практической работы
	Промежуточная аттестация	36					Экзамен
	Итого семестр 3	144	0	54		54	

3.2. Содержание занятий по видам учебной работы

Таблица 6 – Содержание дисциплины

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
Содержание практических работ, 3 семестр		
1.	1. Типы данных. Простые типы данных 1.1. Числовые типы данных 1.2. Логические типы данных 1.3. Другие типы данных	Структура программ на Java. Классы в Java. Пакеты. Правила именования элементов программы. Комментарии в программе. Типы переменных. Примитивные и объектные типы данных. Преобразование типов. Методы. Сигнатура методов. Строки. Массивы. Управляющие конструкции. Обработка исключений. Основные сведения о среде Eclipse. Настройка среды. Разработка, редактирование и отладка приложений в среде. Практическая работа № 1 Работа с примитивными типами языка Java.
2.	2. Типы данных. Структурные типы данных	Практическая работа № 2-4 Строки. Массивы. Управляющие конструкции. Обработка

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
	2.1. Записи 2.2. Массивы 2.3. Строки 2.4. Множества 2.5. Стеки 2.6. Очереди 2.7. Списки 2.8. Деревья	исключений.
3.	3. Понятие алгоритма. 3.1 Свойства алгоритмов 3.1 Классификация алгоритмов по форме представления, по структуре	Практическое занятие № 5-6 Разработка алгоритмов решения задач. Использование стандартных алгоритмов.
4	4. Алгоритмы. Понятие сложности алгоритмов 4.1 Определение сложности алгоритма 4.2 Способы определения сложности алгоритмов 4.3 Сложность рекурсивных алгоритмов. Простая рекурсия 4.4 Сложность рекурсивных алгоритмов. Сложная рекурсия 4.5 Функции, используемые для определения сложности алгоритмов	Практическое занятие № 7-9 Определение сложности алгоритма.

4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.

Для положительной оценки по результатам освоения дисциплины обучающемуся необходимо выполнить все установленные виды учебной работы. Оценка результатов работы обучающегося в баллах (по видам) приведена в таблице 7.

Таблица 7 - Балльно-рейтинговая оценка результатов учебной работы обучающихся по видам (БРС)

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации	Баллы (18 недель)
Текущая учебная работа ОФО (3 семестр)				
Текущая учебная работа в семестре (посещение занятий по расписанию и выполнение заданий)	60 (100% /баллов приведенной шкалы)	Практические (27 работ)	51/27 балл – посещение 1 занятия и выполнение задания на 51-85% 60/27 балла – посещение 1 занятия и существенный вклад на занятии в работу всей группы, самостоятельность и выполнение работы на 85.1-100%	51-60
		Итого по текущей работе в семестре		
Промежуточная аттестация				
Промежуточная аттестация (зачет)	40 (100% /баллов приведенной шкалы)	Вопрос 1.	10баллов (пороговое значение) 20 баллов (максимальное значение)	10 - 20
		Решение задачи 1.	10 баллов (пороговое значение) 20 баллов (максимальное значение)	10 - 20

Итого по промежуточной аттестации (экзамен)	20-40
Суммарная оценка по дисциплине: Сумма баллов текущей и промежуточной аттестации 51 – 100 баллов.	

Итоговая оценка выставляется в ведомость согласно следующему правилу (таблица 8):

Таблица 8 Оценка уровня сформированности компетенций в промежуточной аттестации

Критерии оценивания компетенции	Уровень сформированности компетенции	Итоговая оценка	Оценка по 100-балльной шкале
Обучающийся не владеет теоретическими основами дисциплины и научной терминологией, демонстрирует отрывочные знания, не способен решать практические профессиональные задачи, допускает множественные существенные ошибки в ответах, не умеет интерпретировать результаты и делать выводы.	недопустимый	неудовлетворительно	Менее 51 балла
Обучающийся владеет частично теоретическими основами дисциплины и научной терминологией, фрагментарно способен решать практические профессиональные задачи, допускает несколько существенных ошибок решениях, может частично интерпретировать полученные результаты, допускает ошибки в выводах.	пороговый	удовлетворительно	51-65
Обучающийся владеет теоретическими основами дисциплины и научной терминологией, грамотно излагает материал, способен решать практические профессиональные задачи, но допускает отдельные несущественные ошибки в интерпретации результатов и выводах.	повышенный	хорошо	66-85
Обучающийся в полной мере владеет теоретическими основами дисциплины и научной терминологией, грамотно излагает материал, способен иллюстрировать ответ примерами, фактами, данными научных исследований, применять теоретические знания для решения практических профессиональных задач. Правильно интерпретирует полученные результаты и делает обоснованные выводы.	продвинутый	отлично	86-100

5 Материально-техническое, программное и учебно-методическое обеспечение дисциплины.

5.1 Учебная литература

Основная учебная литература

1. Царев, Р. Ю. Алгоритмы и структуры данных (CDIO): Учебник / Царев Р.Ю., Прокопенко А.В. - Краснояр.:СФУ, 2016. - 204 с.: ISBN 978-5-7638-3388-1. - Текст : электронный. - URL: <https://new.znanium.com/read?id=328418> (дата обращения: 15.03.2020)

Абрамян, М. Э. Практикум по программированию на языке Паскаль: массивы, строки, файлы, рекурсия, линейные динамические структуры, бинарные деревья : учеб. пособие / М. Э. Абрамян. - Ростов н/Д : Издательство ЮФУ, 2010. - 276 с. - ISBN 978-5-9275-0801-3. - Текст : электронный. - URL: <https://znanium.com/catalog/product/549917> (дата обращения: 23.09.2022). – Режим доступа: по подписке.

Дополнительная литература

1. Белов, В. В. Алгоритмы и структуры данных: Учебник / Белов В.В., Чистякова В.И. - Москва :КУРС, НИЦ ИНФРА-М, 2020. - 240 с.: - (Бакалавриат). - ISBN 978-5-16-104748-4. - Текст : электронный. - URL: <https://new.znaniium.com/read?id=347241> (дата обращения: 15.03.2020)
2. Дроздов, С. Н. Структуры и алгоритмы обработки данных: Учебное пособие / Дроздов С.Н. - Таганрог:Южный федеральный университет, 2016. - 228 с.: ISBN 978-5-9275-2242-2. - Текст : электронный. - URL: <https://new.znaniium.com/catalog/product/991928> (дата обращения: 15.03.2020)
3. Монахов, В. В. Язык программирования Java и среда NetBeans: Курс лекций / Монахов В.В. - СПб:БХВ-Петербург, 2011. - 703 с.ISBN 978-5-9775-0671-7. - Текст : электронный. - URL: <https://znaniium.com/catalog/product/355260>
4. Пруцков, А.В. Программирование на языке Java. Введение в курс с примерами и практическими заданиями : учебник / А.В. Пруцков. — М. : КУРС, 2018.- 208 с. - ISBN 978-5-906923-51-6. - Текст : электронный. - URL: <https://new.znaniium.com/read?id=339436> (дата обращения: 28.02.2020)

5.2 Материально-техническое и программное обеспечение дисциплины.

Учебные занятия по дисциплине проводятся в учебных аудиториях НФИ КемГУ:

Таблица 8 – Информационные технологии и программное обеспечение аудиторных занятий и самостоятельной работы

Наименование помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом, в том числе помещения для самостоятельной работы	Перечень основного оборудования, учебно-наглядных пособий и используемого программного обеспечения	Адрес (местоположение) помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом
<p>615 Учебная аудитория (мультимедийная) для проведения:</p> <ul style="list-style-type: none"> - занятий лекционного типа; - занятий семинарского (практического) типа; - текущего контроля и промежуточной аттестации; - государственной итоговой аттестации. 	<p>Специализированная (учебная) мебель: доска меловая, кафедра, столы, стулья.</p> <p>Оборудование для презентации учебного материала: стационарное - компьютер, экран, проектор, акустическая система (колонки).</p> <p>Используемое программное обеспечение: UbuntuLinux(свободно распространяемое ПО), LibreOffice (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО).</p> <p>Интернет с обеспечением доступа в ЭИОС</p>	<p>654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19</p>

<p>502 Компьютерный класс. Учебная аудитория (мультимедийная) для проведения: - занятий семинарского (практического) типа; - групповых и индивидуальных консультаций; - самостоятельной работы.</p>	<p>Специализированная (учебная) мебель: доска меловая, столы компьютерные, стулья. Оборудование для презентации учебного материала: стационарное - компьютер, экран, проектор. Оборудование: стационарное – компьютеры для обучающихся (16 шт.). Используемое программное обеспечение: MSWindows (Microsoft Imagine Premium 3 year по лицензионному договору № 1212/КМР от 12.12.2018 г. до 12.12.2021 г.), LibreOffice (свободно распространяемое ПО), Bloodshed DevC++ 4.9.9.2 (свободно распространяемое ПО), Foxit Reader (свободно распространяемое ПО), Firefox 14 (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО), Java (бесплатная версия), Eclipse (свободно распространяемое ПО), Microsoft Visual Studio (Microsoft Imagine Premium 3 year по лицензионному договору № 1212/КМР от 12.12.2018 г. до 12.12.2021 г.) Интернет с обеспечением доступа в ЭИОС.</p>	<p>654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19</p>
---	---	--

<p>712 Учебная аудитория (мультимедийная) для проведения:</p> <ul style="list-style-type: none"> - групповых и индивидуальных консультаций; - текущего контроля и промежуточной аттестации. 	<p>Специализированная (учебная) мебель: доска меловая, кафедра, столы, стулья.</p> <p>Оборудование для презентации учебного материала: стационарное - компьютер, экран, проектор, акустическая система.</p> <p>Используемое программное обеспечение: MSWindows (Microsoft Imagine Premium 3 year по лицензионному договору № 1212/КМР от 12.12.2018 г. до 12.12.2021 г.), LibreOffice (свободно распространяемое ПО), FoxitReader (свободно распространяемое ПО), Firefox 14 (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО).</p> <p>Интернет с обеспечением доступа в ЭИОС.</p>	<p>654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19</p>
---	--	--

5.3 Современные профессиональные базы данных и информационные справочные системы.

1. CITForum.ru - on-line библиотека свободно доступных материалов по информационным технологиям на русском языке - <http://citforum.ru>

2. Научная электронная библиотека eLIBRARY.RU – крупнейший российский информационный портал в области науки, технологии, медицины и образования, содержащий рефераты и полные тексты - www.elibrary.ru

3. Единое окно доступа к образовательным ресурсам - <http://window.edu.ru/>

6 Иные сведения и (или) материалы.

6.1. Примерные вопросы и задания / задачи для промежуточной аттестации

Таблица 9 - Примерные теоретические вопросы к зачету 1 семестр

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания / задачи
<p>1. Типы данных. Простые типы данных</p> <p>1.1. Числовые типы данных</p> <p>1.2. Логические типы данных</p> <p>1.3. Другие типы данных</p>	<p>1. Могут ли два разных типа данных иметь одинаковое множество значений?</p> <p>2. Могут ли два разных типа данных иметь одинаковое множество операций?</p> <p>3. Какие типы данных относятся к простым?</p> <p>4. Что такое интерфейс и реализация типа данных?</p> <p>5. Что такое сигнатура операции?</p> <p>6. Что такое инкапсуляция данных?</p>	<p>Типовое практическое задание</p>

	<p>7. Что такое интерфейс в языках Java, C#?</p> <p>8. Чем понятие интерфейса отличается от понятия класса?</p> <p>9. В каком случае разность двух беззнаковых целочисленных значений может оказаться больше, чем уменьшаемое?</p> <p>10. Чем отличается вещественный тип с фиксированной точкой от целочисленного типа?</p> <p>11. Почему умножение вещественных чисел с фиксированной точкой отличается от умножения целых чисел необходимостью сдвига результата?</p> <p>12. Можно ли реализовать тип рациональных чисел с неограниченно большими числителем и знаменателем?</p> <p>13. Почему типы данных с повышенной точностью применяются в программировании довольно редко?</p> <p>14. Известно, что в математике результат операции конъюнкции равен 0 тогда и только тогда, когда хотя бы один операнд равен 0. Справедливо ли это утверждение для операций '&&' и '&' языка C?</p> <p>15. Известно, что в математике результат операции дизъюнкции равен 0 тогда и только тогда, когда оба операнда равны 0. Справедливо ли это утверждение для операций ' ' и ' ' языка C?</p> <p>16. В чем преимущества и недостатки того, что в языке C булевы операции можно применять к целочисленным значениям?</p> <p>17. Почему для символьных типов важна алфавитная упорядоченность?</p> <p>18. В чем смысл использования ограниченного типа данных, если соответствующий базовый тип позволяет не менее эффективно реализовать такой же набор значений и операций?</p>	
<p>2. Типы данных. Структурные типы данных</p> <p>2.1. Записи</p> <p>2.2. Массивы</p> <p>2.3. Строки</p> <p>2.4. Множества</p> <p>2.5. Стеки</p> <p>2.6. Очереди</p> <p>2.7. Списки</p> <p>2.8. Деревья</p>	<p>1. В чем заключаются преимущества и недостатки векторного размещения элементов структуры?</p> <p>2. В чем заключаются преимущества и недостатки сцепленного представления структуры?</p> <p>3. Сколько операций сложения и умножения потребуется для вычисления адреса элемента массива с k индексами?</p> <p>4. В чем сходство строк символов с массивами и в чем их различия?</p> <p>5. Почему в языке Pascal можно определить тип множества символов, но нельзя определить тип множества произвольных целых чисел?</p> <p>6. Почему рекурсивное определение обязательно должно содержать нерекурсивную альтернативу?</p> <p>7. Какое представление данных в стеке используется в системах программирования для реализации рекурсивного вызова функций?</p> <p>8. Какое представление стека следует использовать, если максимальное количество элементов в стеке невозможно предсказать заранее?</p> <p>9. Какой недостаток векторного размещения не играет роли при реализации стека в массиве?</p> <p>10. Какой недостаток сцепленного размещения не играет роли при реализации стека в виде сцепленного списка?</p> <p>11. Почему при сцепленном представлении стека в функции PUSH не предусмотрена обработка ошибки?</p> <p>12. Верно ли утверждение: «После присваивания 'p1 := p2' переменная-указатель p1 указывает на p2»?</p> <p>13. Почему при сцепленном представлении очереди указатель должен ссылаться на последний, а не на первый элемент?</p> <p>14. Как можно модифицировать представление очереди в виде сцепленного списка, чтобы можно было эффективно выполнять все операции, определенные для дека?</p> <p>15. Почему представления приоритетной очереди как обычного</p>	<p>Типовое практическое задание</p>

	<p>массива или списка являются неэффективными?</p> <p>16. Иногда неопытные программисты пытаются реализовать просеивание элемента через пирамиду «чуть-чуть» иначе. Вместо сравнения сыновей узла и выбора меньшего из них выполняется</p> <ol style="list-style-type: none"> 1. В чем заключаются преимущества и недостатки векторного размещения элементов структуры? 2. В чем заключаются преимущества и недостатки сцепленного представления структуры? 3. Сколько операций сложения и умножения потребуется для вычисления адреса элемента массива с k индексами? 4. В чем сходство строк символов с массивами и в чем их различия? 5. Почему в языке Pascal можно определить тип множества символов, но нельзя определить тип множества произвольных целых чисел? 6. Почему рекурсивное определение обязательно должно содержать нерекурсивную альтернативу? 7. Какое представление данных в стеке используется в системах программирования для реализации рекурсивного вызова функций? 8. Какое представление стека следует использовать, если максимальное количество элементов в стеке невозможно предсказать заранее? 9. Какой недостаток векторного размещения не играет роли при реализации стека в массиве? 10. Какой недостаток сцепленного размещения не играет роли при реализации стека в виде сцепленного списка? 11. Почему при сцепленном представлении стека в функции PUSH не предусмотрена обработка ошибки? 12. Верно ли утверждение: «После присваивания 'p1 := p2' переменная-указатель p1 указывает на p2»? 13. Почему при сцепленном представлении очереди указатель должен ссылаться на последний, а не на первый элемент? 14. Как можно модифицировать представление очереди в виде сцепленного списка, чтобы можно было эффективно выполнять все операции, определенные для дека? 15. Почему представления приоритетной очереди как обычного массива или списка являются неэффективными? 16. Иногда неопытные программисты пытаются реализовать просеивание элемента через пирамиду «чуть-чуть» иначе. Вместо сравнения сыновей узла и выбора меньшего из них выполняется 	
<p>3. Понятие алгоритма. 3.1 Свойства алгоритмов 3.1 Классификация алгоритмов по форме представления, по структуре</p>	<ol style="list-style-type: none"> 1. Приведите примеры различных видов поиска, используемого при работе компьютерных систем. 2. В чем преимущество эмпирического оценивания эффективности? 3. В чем преимущество теоретического оценивания эффективности? 4. Какой из двух подходов позволяет оценить эффективность алгоритма независимо от его конкретной программной реализации? 5. Что такое вычислительная сложность алгоритма? 6. Почему в оценках типа «O-большое» не указывается основание логарифмов? 7. Почему для небольших массивов данных алгоритм с худшей оценкой может работать быстрее? 8. Приведите примеры практического использования операции сортировки. 9. Пусть дан словарь некоторого языка и требуется отыскать в нем все анаграммы (слова, получаемые одно из другого путем перестановки букв). Как может сортировка помочь при решении этой задачи? 10. Алгоритм «шейкер-сортировки» работает подобно 	<p>Типовое практическое задание</p>

	<p>алгоритму пузырька, но при этом все нечетные проходы выполняются по возрастанию индексов, а четные – по убыванию. Какую оценку времени работы имеет такой алгоритм?</p> <p>11. Почему алгоритм выбора работает быстрее, чем алгоритм пузырька?</p> <p>12. Некоторые программисты пытаются при реализации алгоритма вставок искать место вставки не по убыванию индекса (a_k, a_{k-1}, a_{k-2} и т.д.), а по возрастанию (a_1, a_2, a_3 и т.д.). Чем это плохо?</p> <p>13. Что будет, если для сортировки цепочек в алгоритме Шелла использовать не вставки, а выбор или «пузырек»?</p> <p>14. Зачем в алгоритме Шелла выполняются проходы с $h > 1$, если последний проход (при $h = 1$) наверняка обеспечивает сортировку массива?</p> <p>15. Что такое операция разделения массива в алгоритме QuickSort?</p> <p>16. Какой из двух отрезков, образовавшихся после разделения массива, следует заносить в стек и почему?</p> <p>17. В каком случае выбор первого элемента массива в качестве разделяющего является приемлемым?</p> <p>18. Почему в алгоритме HeapSort просеивание элементов надо начинать с середины массива?</p>	
<p>4. Алгоритмы. Понятие сложности алгоритмов</p> <p>4.1 Определение сложности алгоритма</p> <p>4.2 Способы определения сложности алгоритмов</p> <p>4.3 Сложность рекурсивных алгоритмов. Простая рекурсия</p> <p>4.4 Сложность рекурсивных алгоритмов. Сложная рекурсия</p> <p>4.5 Функции, используемые для определения сложности алгоритмов</p>	<p>1. Что такое сложность задачи?</p> <p>2. Чем отличается понятие сложности задачи от сложности алгоритма?</p> <p>3. В чем заключается трудность определения сложности задачи?</p> <p>4. Что такое массовая задача?</p> <p>5. Что такое длина описания задачи и в каких единицах она измеряется?</p> <p>6. Приведите пример «неразумной» (слишком длинной) кодировки целых чисел.</p> <p>7. Почему в теории вычислительной сложности в качестве средства задания алгоритмов используется машина Тьюринга?</p> <p>8. Что такое задача распознавания?</p> <p>9. Как можно привести задачу оптимизации к задачам распознавания?</p> <p>10. В чем полезность понятия сложности задачи для практики?</p> <p>11. Дайте определение полиномиальной сводимости задачи.</p> <p>12. Чем отличается определение недетерминированной машины Тьюринга от детерминированной?</p> <p>13. Можно ли создать реальное вычислительное устройство, работающее как недетерминированная машина Тьюринга?</p> <p>14. Что означает выражение: «Индивидуальная задача распознавания принимается данной детерминированной машиной Тьюринга»?</p> <p>15. Что означает выражение: «Индивидуальная задача распознавания принимается данной недетерминированной машиной Тьюринга»?</p> <p>16. Что такое класс задач NP?</p> <p>17. Что такое полиномиально проверяемая задача?</p> <p>18. $P = NP$?</p> <p>19. Что такое NP-трудные и NP-полные задачи?</p> <p>20. В чем важность понятия NP-полной задачи?</p> <p>21. Как доказать NP-полноту задачи Z2, опираясь на доказанную NP-полноту задачи Z1?</p> <p>22. Почему и как возможно доказать сводимость бесконечного количества NP-задач к одной конкретной задаче?</p> <p>23. Сформулируйте теорему Кука.</p> <p>24. В чем смысл шести условий, накладываемых на работу НМТ в доказательстве теоремы Кука?</p> <p>25. Как удается свести общую задачу ВВП к более частной</p>	

	<p>задаче 3-ВЫП?</p> <p>26. Придумайте (или найдите в литературе) эффективный алгоритм решения задачи о двух станках.</p> <p>27. Почему задача о рюкзаке, являющаяся NP-полной, допускает эффективное решение с помощью алгоритма Беллмана?</p> <p>28. Что такое псевдополиномиальные задачи?</p> <p>29. Считаете ли вы, что программисту полезно знать основы теории сложности вычислений?</p>	
--	---	--

Типовые практические задания

1. Построить блок-схему линейного алгоритма вычисления арифметической прогрессии
2. Построить блок-схему линейного алгоритма вычисления геометрической прогрессии
3. Определить сложность умножения двух матриц
4. Используя массивы, указатели и структуры, реализуйте вывод и заполнение динамической структуры очередь
5. Реализовать разработанный алгоритм в виде приложения на Java в IDE Eclipse.
6. Реализовать нормальный алгоритм Маркова для инвертирования бинарного числа.

Составитель: Новоселова О.И., ст. препод. кафедры информатики и вычислительной техники им. В.К. Буторина