МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»
(ТУСУР)

УТВЕРЖДАЮ
Проректор по учебной работе
_____П.В. Сенченко
«23» _____12_____ 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ПРОГРАММИРОВАНИЕ

Уровень образования: **высшее образование - бакалавриат**
Направление подготовки / специальность: **09.03.04 Программная инженерия**
Направленность (профиль) / специализация: **Индустриальная разработка программных продуктов**
Форма обучения: **заочная (в том числе с применением дистанционных образовательных технологий)**
Факультет: **Факультет дистанционного обучения (ФДО)**
Кафедра: **Кафедра автоматизации обработки информации (АОИ)**
Курс: **1**
Семестр: **2**
Учебный план набора 2021 года

Объем дисциплины и виды учебной деятельности

Виды учебной деятельности	2 семестр	Всего	Единицы
Лабораторные занятия	8	8	часов
Самостоятельная работа	213	213	часов
Самостоятельная работа под руководством преподавателя	20	20	часов
Контрольные работы	2	2	часов
Подготовка и сдача экзамена	9	9	часов
Общая трудоемкость (включая промежуточную аттестацию)	252	252	часов
		7	з.е.

Формы промежуточной аттестация	Семестр	Количество
Экзамен	2	
Контрольные работы	2	1

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Сенченко П.В.
Должность: Проректор по УР
Дата подписания: 23.12.2020
Уникальный программный ключ:
a1119608-cdff-4455-b54e-5235117c185c

Томск

Согласована на портале № 73817

1. Общие положения

1.1. Цели дисциплины

1. Формирование навыков алгоритмизации и структурного программирования.

1.2. Задачи дисциплины

1. Обучение студентов навыкам разработки и отладки программ на алгоритмических языках программирования.
2. Изучение техники объектно-ориентированного анализа.
3. Изучение приемов объектно-ориентированного программирования.

2. Место дисциплины в структуре ОПОП

Блок дисциплин: Б1. Дисциплины (модули).

Часть блока дисциплин: Обязательная часть.

Модуль дисциплин: Модуль укрупненной группы специальностей и направлений (general hard skills – GHS).

Индекс дисциплины: Б1.О.02.03.

Реализуется с применением электронного обучения, дистанционных образовательных технологий.

3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Процесс изучения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО и основной образовательной программой (таблица 3.1):

Таблица 3.1 – Компетенции и индикаторы их достижения

Компетенция	Индикаторы достижения компетенции	Планируемые результаты обучения по дисциплине
Универсальные компетенции		
-	-	-
Общепрофессиональные компетенции		

Согласована на портале № 73817

ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности	ОПК-2.1. Знает основы информационных технологий и программирования; основные компоненты программных средств, а также их назначение и состав	Знает методы структурного программирования и основные компоненты программы, написанной в стиле структурного программирования
	ОПК-2.2. Умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности, а также обосновывать их выбор	Умеет выбирать среды программирования для решения задач профессиональной деятельности, а также обосновывать их выбор
	ОПК-2.3. Владеет навыками применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	Владеет навыками применения языка программирования Си при решении задач профессиональной деятельности
Профессиональные компетенции		
-	-	-

4. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 7 зачетных единиц, 252 академических часов.

Распределение трудоемкости дисциплины по видам учебной деятельности представлено в таблице 4.1.

Таблица 4.1 – Трудоемкость дисциплины по видам учебной деятельности

Виды учебной деятельности	Всего часов	Семестры
		2 семестр
Контактная аудиторная работа обучающихся с преподавателем, всего	30	30
Лабораторные занятия	8	8
Самостоятельная работа под руководством преподавателя	20	20
Контрольные работы	2	2
Самостоятельная работа обучающихся, в т.ч. контактная внеаудиторная работа обучающихся с преподавателем, всего	213	213
Самостоятельное изучение тем (вопросов) теоретической части дисциплины	175	175
Подготовка к контрольной работе	6	6
Подготовка к лабораторной работе	16	16
Написание отчета по лабораторной работе	16	16

Согласована на портале № 73817

Подготовка и сдача экзамена	9	9
Общая трудоемкость (в часах)	252	252
Общая трудоемкость (в з.е.)	7	7

5. Структура и содержание дисциплины

5.1. Разделы (темы) дисциплины и виды учебной деятельности

Структура дисциплины по разделам (темам) и видам учебной деятельности приведена в таблице 5.1.

Таблица 5.1 – Разделы (темы) дисциплины и виды учебной деятельности

Названия разделов (тем) дисциплины	Лаб. раб.	Контр. раб.	СРП, ч.	Сам. раб., ч	Всего часов (без промежуточной аттестации)	Формируемые компетенции
2 семестр						
1 Основы алгоритмизации	-	2	4	56	62	ОПК-2
2 Сортировка и поиск	-		6	58	64	ОПК-2
3 Численные алгоритмы	8		10	99	117	ОПК-2
Итого за семестр	8	2	20	213	243	
Итого	8	2	20	213	243	

5.2. Содержание разделов (тем) дисциплины

Содержание разделов (тем) дисциплины приведено в таблице 5.2.

Таблица 5.2 – Содержание разделов (тем) дисциплины

Названия разделов (тем) дисциплины	Содержание разделов (тем) дисциплины	СРП, ч	Формируемые компетенции
2 семестр			
1 Основы алгоритмизации	Вложенные и параллельные алгоритмы. Логические элементы и базовые управляющие структуры визуального структурного программирования. Построение алгоритма из базовых структур. Пошаговая детализация как метод проектирования алгоритмов	4	ОПК-2
	Итого	4	
2 Сортировка и поиск	Алгоритмы поиска: последовательный, бинарный. Алгоритмы сортировки элементов массива: простой выбор, простая замена, простой обмен. Усовершенствованные методы сортировки.	6	ОПК-2
	Итого	6	
3 Численные алгоритмы	Решение систем линейных уравнений метод Гаусса. Обусловленность системы линейных алгебраических уравнений. Большие разреженные системы. Полиномиальная и сплайн-интерполяция. Численное интегрирование методами прямоугольников, трапеция и Симпсона. Решение нелинейных уравнений. Численное решение задачи Коши для обыкновенных дифференциальных уравнений.	10	ОПК-2
	Итого	10	

Согласована на портале № 73817

Итого за семестр	20	
Итого	20	

5.3. Контрольные работы

Виды контрольных работ и часы на контрольные работы приведены в таблице 5.3.

Таблица 5.3 – Контрольные работы

№ п.п.	Виды контрольных работ	Трудоемкость, ч	Формируемые компетенции
2 семестр			
1	Контрольная работа с автоматизированной проверкой	2	ОПК-2
Итого за семестр		2	
Итого		2	

5.4. Лабораторные занятия

Наименование лабораторных работ приведено в таблице 5.4.

Таблица 5.4 – Наименование лабораторных работ

Названия разделов (тем) дисциплины	Наименование лабораторных работ	Трудоемкость, ч	Формируемые компетенции
2 семестр			
3 Численные алгоритмы	Решение систем линейных алгебраических уравнений методом Гаусса. Численное интегрирование методами центральных прямоугольников, трапеций и Симпсона.	4	ОПК-2
	Численное решение задачи Коши. Решение нелинейного уравнения	4	ОПК-2
	Итого	8	
Итого за семестр		8	
Итого		8	

5.5.Контроль самостоятельной работы (курсовой проект / курсовая работа)

Не предусмотрено учебным планом

5.6. Самостоятельная работа

Виды самостоятельной работы, трудоемкость и формируемые компетенции представлены в таблице 5.6.

Таблица 5.6. – Виды самостоятельной работы, трудоемкость и формируемые компетенции

Названия разделов (тем) дисциплины	Виды самостоятельной работы	Трудоемкость, ч	Формируемые компетенции	Формы контроля
2 семестр				

Согласована на портале № 73817

1 Основы алгоритмизации	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	54	ОПК-2	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-2	Контрольная работа
	Итого	56		
2 Сортировка и поиск	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	56	ОПК-2	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-2	Контрольная работа
	Итого	58		
3 Численные алгоритмы	Подготовка к лабораторной работе	16	ОПК-2	Лабораторная работа
	Написание отчета по лабораторной работе	16	ОПК-2	Отчет по лабораторной работе
	Самостоятельное изучение тем (вопросов) теоретической части дисциплины	65	ОПК-2	Тестирование, Экзамен
	Подготовка к контрольной работе	2	ОПК-2	Контрольная работа
	Итого	99		
Итого за семестр		213		
	Подготовка и сдача экзамена	9		Экзамен
Итого		222		

5.7. Соответствие компетенций, формируемых при изучении дисциплины, и видов учебной деятельности

Соответствие компетенций, формируемых при изучении дисциплины, и видов учебной деятельности представлено в таблице 5.7.

Таблица 5.7 – Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Формируемые компетенции	Виды учебной деятельности				Формы контроля
	Лаб. раб.	Конт.Раб.	СРП	Сам. раб.	
ОПК-2	+	+	+	+	Контрольная работа, Лабораторная работа, Отчет по лабораторной работе, Тестирование, Экзамен

6. Рейтинговая система для оценки успеваемости обучающихся

Рейтинговая система не используется

7. Учебно-методическое и информационное обеспечение дисциплины

7.1. Основная литература

Согласована на портале № 73817

1. Артёмов И. Л. Информатика III : учебное пособие / И. Л. Артёмов. – Томск : ФДО, ТУСУР, 2018. – 95 с. Доступ из личного кабинета студента. [Электронный ресурс]: — Режим доступа: https://study.tusur.ru/study/library/.

7.2. Дополнительная литература

1. Трофимов, В. В. Алгоритмизация и программирование : учебник для вузов / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2020. — 137 с. Доступ из личного кабинета студента. [Электронный ресурс]: — Режим доступа: https://urait.ru/bcode/452333.
2. Кувшинов, Д. Р. Основы программирования : учебное пособие для вузов / Д. Р. Кувшинов. — Москва : Издательство Юрайт, 2020. — 104 с. Доступ из личного кабинета студента. [Электронный ресурс]: — Режим доступа: https://urait.ru/bcode/454667.

7.3. Учебно-методические пособия

7.3.1. Обязательные учебно-методические пособия

1. Пермякова Н. В. Информатика и программирование : методические указания по организации самостоятельной работы для студентов заочной формы обучения технических направлений подготовки, обучающихся с применением дистанционных образовательных технологий / Н. В. Пермякова, Ю.В. Морозова. – Томск : ФДО, ТУСУР, 2018. – 22 с. Доступ из личного кабинета студента. [Электронный ресурс]: — Режим доступа: https://study.tusur.ru/study/library/.
2. Артёмов И. Л. Информатика III : методические указания по выполнению лабораторных работ / И. Л. Артёмов. – Томск : ФДО, ТУСУР, 2018. – 48 с. Доступ из личного кабинета студента. [Электронный ресурс]: — Режим доступа: https://study.tusur.ru/study/library/.

7.3.2. Учебно-методические пособия для лиц с ограниченными возможностями здоровья и инвалидов

Учебно-методические материалы для самостоятельной и аудиторной работы обучающихся из числа лиц с ограниченными возможностями здоровья и инвалидов предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации.

Для лиц с нарушениями зрения:
– в форме электронного документа;
– в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:
– в форме электронного документа;
– в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:
– в форме электронного документа;
– в печатной форме.

7.4. Иное учебно-методическое обеспечение

1. Артёмов И. Л. Информатика III [Электронный ресурс]: электронный курс / И. Л. Артёмов. – Томск :ФДО, ТУСУР, 2018. (доступ из личного кабинета студента) .

7.5. Современные профессиональные базы данных и информационные справочные системы

1. При изучении дисциплины рекомендуется обращаться к современным базам данных, информационно-справочным и поисковым системам, к которым у ТУСУРа открыт доступ: https://lib.tusur.ru/ru/resursy/bazy-dannyh.

8. Материально-техническое и программное обеспечение дисциплины

8.1. Общие требования к материально-техническому и программному обеспечению дисциплины

Учебные аудитории для проведения занятий лабораторного типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, для

Согласована на портале № 73817

самостоятельной работы студентов

634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Описание имеющегося оборудования:

- Веб-камера - 6 шт.;
- Наушники с микрофоном - 6 шт.;
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

- 7-Zip;
- DEV C++ (с возможностью удаленного доступа);
- Google Chrome;
- Kaspersky Endpoint Security для Windows;
- LibreOffice 7.0.6.2;
- Microsoft Windows;
- Notepad++ (с возможностью удаленного доступа);

8.2. Материально-техническое и программное обеспечение для самостоятельной работы

Для самостоятельной работы используются учебные аудитории (компьютерные классы), расположенные по адресам:

- 634050, Томская область, г. Томск, Ленина проспект, д. 40, 233 ауд.;
- 634045, Томская область, г. Томск, ул. Красноармейская, д. 146, 209 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 47, 126 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Описание имеющегося оборудования:

- учебная мебель;
- компьютеры;
- компьютеры подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду ТУСУРа.

Перечень программного обеспечения:

- Microsoft Windows;
- OpenOffice;
- Kaspersky Endpoint Security 10 для Windows;
- 7-Zip;
- Google Chrome.

8.3. Материально-техническое обеспечение дисциплины для лиц с ограниченными возможностями здоровья и инвалидов

Освоение дисциплины лицами с ограниченными возможностями здоровья и инвалидами осуществляется с использованием средств обучения общего и специального назначения.

При занятиях с обучающимися с **нарушениями слуха** предусмотрено использование звукоусиливающей аппаратуры, мультимедийных средств и других технических средств приема/передачи учебной информации в доступных формах, мобильной системы преподавания для обучающихся с инвалидностью, портативной индукционной системы. Учебная аудитория, в которой занимаются обучающиеся с нарушением слуха, оборудована компьютерной техникой, аудиотехникой, видеотехникой, электронной доской, мультимедийной системой.

При занятиях с обучающимися с **нарушениями зрения** предусмотрено использование в лекционных и учебных аудиториях возможности просмотра удаленных объектов (например, текста на доске или слайда на экране) при помощи видеоувеличителей для комфортного просмотра.

При занятиях с обучающимися с **нарушениями опорно-двигательного аппарата** используются альтернативные устройства ввода информации и другие технические средства приема/передачи учебной информации в доступных формах, мобильной системы обучения для людей с инвалидностью.

9. Оценочные материалы и методические рекомендации по организации изучения дисциплины

Согласована на портале № 73817

9.1. Содержание оценочных материалов для текущего контроля и промежуточной аттестации

Для оценки степени сформированности и уровня освоения закрепленных за дисциплиной компетенций используются оценочные материалы, представленные в таблице 9.1.

Таблица 9.1 – Формы контроля и оценочные материалы

Названия разделов (тем) дисциплины	Формируемые компетенции	Формы контроля	Оценочные материалы (ОМ)
1 Основы алгоритмизации	ОПК-2	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
2 Сортировка и поиск	ОПК-2	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
3 Численные алгоритмы	ОПК-2	Контрольная работа	Примерный перечень тем и тестовых заданий на контрольные работы
		Лабораторная работа	Темы лабораторных работ
		Тестирование	Примерный перечень тестовых заданий
		Экзамен	Перечень экзаменационных вопросов
		Отчет по лабораторной работе	Темы лабораторных работ

Шкала оценки сформированности отдельных планируемых результатов обучения по дисциплине приведена в таблице 9.2.

Таблица 9.2 – Шкала оценки сформированности планируемых результатов обучения по дисциплине

Оценка	Баллы за ОМ	Формулировка требований к степени сформированности планируемых результатов обучения		
		знать	уметь	владеть
2 (неудовлетворительно)	< 60% от максимальной суммы баллов	отсутствие знаний или фрагментарные знания	отсутствие умений или частично освоенное умение	отсутствие навыков или фрагментарные применение навыков

Согласована на портале № 73817

3 (удовлетворительно)	от 60% до 69% от максимальной суммы баллов	общие, но не структурированные знания	в целом успешно, но не систематически осуществляемое умение	в целом успешное, но не систематическое применение навыков
4 (хорошо)	от 70% до 89% от максимальной суммы баллов	сформированные, но содержащие отдельные проблемы знания	в целом успешное, но содержащие отдельные пробелы умение	в целом успешное, но содержащие отдельные пробелы применение навыков
5 (отлично)	≥ 90% от максимальной суммы баллов	сформированные систематические знания	сформированное умение	успешное и систематическое применение навыков

Шкала комплексной оценки сформированности компетенций приведена в таблице 9.3.

Таблица 9.3 – Шкала комплексной оценки сформированности компетенций

Оценка	Формулировка требований к степени компетенции
2 (неудовлетворительно)	Не имеет необходимых представлений о проверяемом материале или Знать на уровне **ориентирования**, представлений. Обучающийся знает основные признаки или термины изучаемого элемента содержания, их отнесенность к определенной науке, отрасли или объектам, узнает в текстах, изображениях или схемах и знает, к каким источникам нужно обращаться для более детального его усвоения.
3 (удовлетворительно)	Знать и уметь на **репродуктивном** уровне. Обучающихся знает изученный элемент содержания репродуктивно: произвольно воспроизводит свои знания устно, письменно или в демонстрируемых действиях.
4 (хорошо)	Знать, уметь, владеть на **аналитическом** уровне. Зная на репродуктивном уровне, указывать на особенности и взаимосвязи изученных объектов, на их достоинства, ограничения, историю и перспективы развития и особенности для разных объектов усвоения.
5 (отлично)	Знать, уметь, владеть на **системном** уровне. Обучающийся знает изученный элемент содержания системно, произвольно и доказательно воспроизводит свои знания устно, письменно или в демонстрируемых действиях, учитывая и указывая связи и зависимости между этим элементом и другими элементами содержания дисциплины, его значимость в содержании дисциплины.

9.1.1. Примерный перечень тестовых заданий

1. Для обработки имеющейся информации была написана программа, соответствующая принципам структурного программирования. Какой тип передачи управления может использовать такая программа?
   1 безусловный
   2 объектно-зависимый
   3 условный
   4 функционально-зависимый

2. Управление информацией может быть организовано с использованием различных

Согласована на портале № 73817

программных
средств, одним из которых являются языки программирования. Какая из перечисленных констант записана верно с точки зрения синтаксиса языка программирования Си?
1. 5,025
2. -12e-0.12
3. 0197
4. 5.

3. Для управления информацией, хранящейся в цифровом виде можно использовать самостоятельно
разработанные программные средства. Процесс разработки таких средств подразумевает владение языком программирования. Какое имя идентификатора переменной, записано неверно,
с точки зрения синтаксиса языка программирования Си?
1. 2a
2. func
3. a_b
4. A2

4. Информация, хранящаяся на компьютере, может обрабатываться в оперативной памяти. Язык программирования Си допускает в явном виде работу с адресами физической памяти – указателями.
Какая из перечисленных переменных описана как указатель?
1. int p[25];
2. int * f;
3. int z[12][3];
4. int &a;

5. Информация, хранящаяся на компьютере, может обрабатываться в оперативной памяти. Язык программирования Си допускает в явном виде работу с адресами физической памяти.
После выполнения какого из перечисленных фрагментов кода в переменной x будет храниться
адрес переменной y?
1. int *x;
int y = 15;
x = *y;
---
2. int *x;
int y = 15;
x = y*;
---
3. int *x;
int y = 15;
x = &y;
---
4. int *x;
int y = 15;
x = #y;
---

6. Для обработки имеющейся информации была написана программа, соответствующая принципам структурного программирования, фрагмент кода которой приведен ниже.
int x = 9;
int y = 4;
int z = 2;
if (x==y && z>0) {z = x;

Согласована на портале № 73817

```
x = y;
y = z;}
else { z = x*2;
x = 2*y;
y = z;}
printf (“ %d %d %d”, x,y,z);
```

Что будет выведено на экран выполнении этого фрагмента?

1. 18 8 18
2. 9 4 9
3. 4 9 9
4. 8 18 18

7. Для обработки имеющейся информации была написана программа, соответствующая принципам структурного программирования, фрагмент кода которой приведен ниже. Что будет выведено
на экран при выполнении этого фрагмента?

```
int i = 25;
do{
printf("%3d",i);
i-=2;
}
while(i>=13);
```

1. 25 23 21 19 17 15 13
2. 25 23 21 19 17 15
3. 13 15 17 19 21 23 25
4. 13 15 17 19 21 23

8. Перед программистом была поставлена задача получения числовой информации в виде последовательности:
2 5 8 11 14 17 20.
Какой из представленных фрагментов решает поставленную задачу?

1.
```
int i = 2;
do{
printf("%3d",i);
i+=3;
}
while(i<20);
```

2.
```
int i = 2;
while(i<=20){
printf("%3d",i);
i+=3;
}
```

3.
```
int i = 2;
while(i>=20){
printf("%3d",i);
i+=3;
}
```

4.
```
int i = 2;
do{
printf("%3d",i);
i+=3;
}
while(i<25);
```

Согласована на портале № 73817

9. Перед программистом поставлена задача получения числовой информации в виде последовательности:
10.0000 5.0000 2.5000 1.2500 0.625 0.3125?
Какой из представленных циклов может быть использован для решения этой задачи?
1. ----------
```
float i;
for (i=10; i>0.2; i-=5)
printf("%.4f ", i);
```
2. ----------
```
float i;
for (i=10; i>0.2; i--)
printf("%.4f ", i);
```
3. ----------
```
float i;
for (i=10; i>0.2; i/=2)
printf("%.4f ", i);
```
4. ----------
```
float i;
for (i=0.3125; i<11; i*=2)
printf("%.4f ", i);
```
---

10. Для решения некоторой задачи обработки массива была использована программа, представленная
ниже:
```
int main(int argc, char *argv[])
{
system("chcp 1251");
int x[10] = {2,7,6,1,9,5,8,3,4,0};
int k = 0,i;
for (i=0;i<10;i++)
if (x[i]%2==0) printf("%3d",i);
printf("\n");
system("PAUSE");
return EXIT_SUCCESS;
}
```
Какую задачу решает написанная программа?
1 Печать индексов нечетных по значению элементов массива
2 Поиск суммы четных по значению элементов массива
3 Поиск суммы нечетных по значению элементов массива
4 Печать индексов четных по значению элементов массива

11. Компьютерная программа, код которой представлен ниже, была использована для обработки
числового массива. Что будет выведено на экран при выполнении программы?
```
int main(int argc, char *argv[])
{
system("chcp 1251");
int x[10] = {6, 9, 2, 3, 7,
7, 6, 11, 8, 5};
int k = x[0],i;
for (i=1;i<10;i++)
if (x[i]>k) k = x[i];
printf("%3d",k);
printf("\n");
system("PAUSE");
```

Согласована на портале № 73817

```
return EXIT_SUCCESS;
}
```
1. 11
2. 2
3. 7
4. 5

12. Компьютерная программа, код которой представлен ниже, была использована для обработки
числового массива. В каком порядке будут расположены элементы массива после выполнения
программы?

```
int main(int argc, char *argv[])
{
system("chcp 1251");
int x[10] = {6, 9, 2, 3, 7,
7, 6, 11, 8, 5};
int i,j,k;
int m = 3;
for (j=0;j<m;j++){
k = x[9];
for (i=9;i>0;i--)
x[i] = x[i-1];
x[0] = k;
}
for (i=0;i<10;i++)
printf("%3d",x[i]);
printf("\n");
system("PAUSE");
return EXIT_SUCCESS;}
```
1. 6 9 2 3 7 7 6 11 8 5
2. 11 8 5 6 9 2 3 7 7 6
3. 8 5 6 9 2 3 7 7 6 11
4. 5 6 9 2 3 7 7 6 11 8

13. Компьютерная программа, код которой представлен ниже, была использована для обработки
числового двумерного массива. Что будет выведено на экран при выполнении программы?

```
int main(int argc, char *argv[])
{
int A[5][5]={1,2,3,4,5,
6,7,8,9,10,
11,12,13,14,15,
16,17,18,19,20,
21,22,23,24,25};
int i,j,k,n=5;
k = A[1][0];
for(i=0;i<n;i++)
for(j=0;j<i;j++)
if (A[i][j]>k) k=A[i][j];
printf("%d\n",k);
system("PAUSE");
return 0;
}
```
1. 6
2. 10

Согласована на портале № 73817

3. 24
4. 25

14. Обрабатываемая в задаче информация была представлена в виде класса языка Си. Что произойдет при компиляции и выполнении?
```
class A {
int b;
public:
A(int i) {b = i;}
A() {b = 111;}
void Show(){ cout<<b;}
}
void main(){
A* obj = new A();
obj ->show();
}
```
1 напечатается 1
2 напечатается 111
3 ошибка компиляции
4 ошибка выполнения

15. При обработке информации данные были описаны с помощью классов языка Си. Каким атрибутом доступа должен быть описан элемент класса, который можно использовать только внутри
класса?
1 public
2 static
3 private
4 protected

16. При обработке информации данные были описаны с помощью классов языка Си. Какое из следующих утверждений является верным?
1 В классе может быть только один конструктор
2 В классе может быть только один деструктор
3 В классе обязательно должен быть написан хотя бы один конструктор
4 Конструктор класса вызывается при каждом обращении к объекту

17. Обрабатываемая в задаче информация была представлена в виде класса языка Си. Что произойдет при компиляции и выполнении?
```
class A {
int b;
public:
A(int i) {b = i;}
A() {b = 0;}
void Show(){ cout<<b;}
}
void main(){
A* obj = new A(1);
cout <<obj->b;
}
```
1 напечатается 0
2 напечатается 1
3 ошибка компиляции
4 ошибка выполнения

18. При обработке информации данные были описаны с помощью классов языка Си. Каким атрибутом доступа должен быть описан элемент класса, который можно использовать

Согласована на портале № 73817

извне класса?
1 public
2 static
3 private
4 protected

19. При обработке информации данные были описаны с помощью классов языка Си. Какое из следующих утверждений является верным?
1 Если в классе нет описания конструктора, то создается конструктор по умолчанию
2 Статические поля могут принимать значения, разные для каждого из объектов класса
3 Конструктор класса обязательно должен быть описан атрибутом доступа public
4 В классе может быть описано несколько конструкторов с одинаковыми списками параметров

20. При обработке информации данные были описаны с помощью класса языка Си. class Point{
private:
int x;
int y;
public:
Point (int x1, int y1);
Point();
void Show();
int GetX();
int GetY();
void PutX(int X);
void PutY(int Y);
};
Какой из перечисленных ниже элементов класса является конструктором класса:
1 int x;
2 Point(int x1, int x2);
3 void Show();
4 int GetX();

9.1.2. Перечень экзаменационных вопросов

Приведены примеры типовых заданий из банка экзаменационных тестов, составленных по пройденным
разделам дисциплины.

1. Какую задачу решает представленный алгоритм?
алг Поиск нач
дано цел X[10]
цел n=10
цел a:=X[1]
для i от 2 до n нц
если X[i] < a то a:=X[i]
кц
рез a
кон
1. Поиск минимального элемента массива.
2. Поиск индекса первого минимального элемента массива.
3. Поиск индекса последнего минимального элемента массива.
4. Поиск максимального элемента массива.

2. Какую задачу решает представленный алгоритм?
алг Поиск нач
дано цел X[10]
цел n=10

Согласована на портале № 73817

цел b:=1
для i от 2 до n нц
если X[i]<X[b] то b:=I
кц
рез b
кон
1. Поиск минимального элемента массива.
2. Поиск индекса первого минимального элемента массива.
3. Поиск индекса последнего минимального элемента массива.
4. Поиск максимального элемента массива.

3. Какую задачу решает представленный алгоритм?
алг Поиск нач
дано цел X[10]
цел n=10
цел b:=1
для i от 2 до n нц
если X[i] ≤ X[b] то b:=I
кц
рез b
кон
1. Поиск минимального элемента массива.
2. Поиск индекса первого минимального элемента массива.
3. Поиск индекса последнего минимального элемента массива.
4. Поиск максимального элемента массива.

4. Какую задачу решает представленный алгоритм?
алг Вопрос4 нач
дано цел X[10]
цел n=10
цел a:=0
для i от 1 до n нц
если X[i] > 0 то a:=a+X[i]
кц
рез a
кон
1. Поиск суммы положительных элементов.
2. Поиск суммы индексов положительных элементов.
3. Поиск количества положительных элементов.
4. Поиск суммы отрицательных элементов.

5. Какую задачу решает представленный алгоритм?
алг Вопрос5 нач
дано цел X[10]
цел n=10
цел a:=0
для i от 1 до n нц
если X[i] > 0 то a:=a+i
кц
рез a
кон
1. Поиск суммы положительных элементов.
2. Поиск суммы индексов положительных элементов.
3. Поиск количества положительных элементов.
4. Поиск суммы отрицательных элементов.

6. Программисту требовалось написать алгоритм поиска индекса первого элемента массива,

Согласована на портале № 73817

имеющего положительное значение. Программист записал алгоритм следующим образом.
1. цел таб A[n]
2. для i от 1 до n нц
3. ввод A[i]
4. кц
5. для i от 1 до n нц
6. если A[i]>0 то p:=I
7. i:=n
8. кц
9. рез p
1. Алгоритм написан верно.
2. В строке 5.
3. В строке 6.
4. В строке 7.

7. Программисту требовалось написать алгоритм поиска индекса первого элемента массива, имеющего положительное значение. Программист записал алгоритм следующим образом.
1. цел таб A[n]
2. для i от 1 до n нц
3. ввод A[i]
4. кц
5. для i от 1 до n нц
6. если A[i]>0 то p:=I
7. i:=i+1
8. кц
9. рез p
1. Алгоритм написан верно.
2. В строке 5.
3. В строке 6.
4. В строке 7..

8. Чему будет равно значение переменной p после выполнения алгоритма, если элементы массива
X ={1, 2, 3, 4, 5, 6, 7, 8, 9, 10} ?
n=10
цел таб X[n]
для i от 1 до n нц
ввод X[i]
кц
p:=1
для i от 2 до n нц
если X[p] > X[i] то p :=i
кц
рез p
1. Алгоритм находит индекс первого минимального элемента.
2. Алгоритм находит индекс последнего минимального элемента.
3. Алгоритм находит индекс первого максимального элемента.
4. Алгоритм находит индекс последнего максимального элемента.

9. Чему будет равно значение переменной p после выполнения алгоритма, если элементы массива
X ={-1, -2, -3, -4, -5, -6, -7, -8, -9, -10} ?
n=10
цел таб X[n]
для i от 1 до n нц
ввод X[i]
кц

Согласована на портале № 73817

p:=1
для i от 2 до n нц
если X[p] >= X[i] то p :=i
кц
рез p
1. Алгоритм находит индекс первого минимального элемента.
2. Алгоритм находит индекс последнего минимального элемента.
3. Алгоритм находит индекс первого максимального элемента.
4. Алгоритм находит индекс последнего максимального элемента.

10. Укажите описание соответствующее фрагменту программы:
int X[10];
int i;
for (i=0;i<10;i++)
X[i]=i+2;
1. Описание и инициализация статического целочисленного массива из 10-ти элементов.
2. Описание и инициализация динамического вещественного массива из 10 –ти элементов.
3. Описание и инициализация статического вещественного массива из 10-ти элементов.
4. Описание и инициализация динамического целочисленного массива из 10 –ти элементов.

9.1.3. Примерный перечень тем и тестовых заданий на контрольные работы

Информатика и программирование

1. Что будет выведено на экран при выполнении следующего фрагмента программы
int i;
for(i=1;i<10;i+=2) printf("%d ",i);
1. Вывод на экран чисел 1, 3, 5, 7, 9
2. Вывод на экран чисел 1, 4, 9, 16, 25
3. Вывести на экран числа 3,5,7,9,11
4. Вывод на экран чисел 3,5,7,9,11

2. Что будет выведено на экран при выполнении следующего фрагмента программы
int i;
for(i=0;i<5;i++) printf("%d ",(i+1)*(i+1));
1. Вывод на экран чисел 1, 3, 5, 7, 9
2. Вывод на экран чисел 1, 4, 9, 16, 25
3. Вывести на экран числа 3,5,7,9,11
4. Вывод на экран чисел 3,5,7,9,11

3. Выберите фрагменты программ, в которых цикл выполняется 7 раз.
1. int i=1; while (i<20) i+=3;
2. int i=0; while (i<15)i+=3;
3. int i=0; while (i<12)i+=2;
4. int i=4; while (i<40) i+=i;

4. Что будет выведено на экран при выполнении следующего фрагмента программы при n = 2500?
int func1(int p){
int s=0;
while(p>0){s+=p%10;
p/=10;
}
return s; }
int main(int argc, char *argv[])

Согласована на портале № 73817

```
{ int n;
scanf("%d",&n);
printf("%d\n",func1(n));
system("PAUSE");
return 0;}
```

1. На экран выведется сумма цифр числа.
2. На экран выведется максимальная цифра числа.
3. На экран выведется минимальная цифра числа.
4. На экран выведется количество цифр числа.

5. Что будет выведено на экран при выполнении следующего фрагмента программы при n = 1000 , m = 5?

```
int func13(int p, int v){
int s=0,t;
while(p>0){t=p%10;
if(t==v)s++;
p/=10;
}
return s; }
int main(int argc, char *argv[])
{ int n,m;
scanf("%d%d",&n,&m);
printf("%d\n",func13(n,m));
system("PAUSE");
return 0;}
```

1. На экран выведется количество цифр числа n равных m.
2. На экран выведется количество цифр числа n не равных m.
3. На экран выведется количество цифр числа n больших m.
4. На экран выведется количество цифр числа n меньших m.

6. Что будет выведено на экран при выполнении следующего фрагмента программы при n = 2020 , m =7?

```
int func16(int p, int v){
int s=0,t;
while(p>0){t=p%10;
if(t<v)s++;
p/=10;
}
return s; }
int main(int argc, char *argv[])
{ int n,m;
scanf("%d%d",&n,&m);
printf("%d\n",func16(n,m));
system("PAUSE");
return 0;}
```

1. На экран выведется количество цифр числа n равных m.
2. На экран выведется количество цифр числа n не равных m.
3. На экран выведется количество цифр числа n больших m.
4. На экран выведется количество цифр числа n меньших m.

7. Чему будет равно значение переменной p при выполнении следующей программы, если элементы
массива y заданы следующим образом y = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}?

```
int Function1(int *x, int f, int l, int z) {
int S = 0,i;
for(i=f;i<l;i+=z) S+=x[i];
return S; }
```

Согласована на портале № 73817

```
int main(int argc, char *argv[])
{ int y[10],i,p;
for(i=0;i<10;i++)
y[i]=rand()%11-rand()%11;
p = Function1(y,0,10,1);
return 0; }
```
1. p - сумма всех элементов массива y.
2. p - сумма первых 5-ти элементов массива.
3. p - сумма элементов массива с четными индексами.
4. p - сумма элементов массива с нечетными индексами.

8. Чему будет равно значение переменной p при выполнении следующей программы, если элементы
массива y заданы следующим образом y = {-10, 2, 3, 4, 5, 6, 7, 8, 9, 1}?
```
p = Function1(y,1,20,2);
int Function1(int *x, int f, int l, int z) {
int S = 0,i;
for(i=f;i<l;i+=z) S+=x[i];
return S; }
int main(int argc, char *argv[])
{ int y[10],i,p;
for(i=0;i<10;i++)
y[i]=rand()%11-rand()%11;
p = Function1(y,1,10,2);
return 0;}
```
1. p - сумма всех элементов массива y.
2. p - сумма первых 5-ти элементов массива.
3. p - сумма элементов массива с четными индексами.
4. p - сумма элементов массива с нечетными индексами.

9. Чему будет равно значение переменной p при выполнении следующей программы, если элементы
массива y заданы следующим образом y = {-10, 2, 3, 4, 5, 6, 7, 8, 9, 1}?
```
p = Function1(y,1,20,2);
int Function1(int *x, int f, int l, int z) {
int S = 0,i;
for(i=f;i<l;i+=z) S+=x[i];
return S; }
int main(int argc, char *argv[])
{ int y[10],i,p;
for(i=0;i<10;i++)
y[i]=rand()%11-rand()%11;
p = Function1(y,1,10,2);
return 0;}
```
1. p - сумма всех элементов массива y.
2. p - сумма первых 5-ти элементов массива.
3. p - сумма элементов массива с четными индексами.
4. p - сумма элементов массива с нечетными индексами.

10. Сколько шагов выполнит внешний шаг сортировки обменом при сортировке элементов массива
X = {-5, 3, 0, -7, 10, -3, -5, 1, 2, 5}?
```
int main() {
int n = 10, X[n],i,C=0,p=0,v,j;
srand(time(NULL));
for(i=0;i<10;i++){
X[i]=rand()%11-rand()%11;
```

Согласована на портале № 73817

```
printf("%d ",X[i]);}
for(i=0;i<n-1;i++){
p=0;C++;
for(j=0;j<n-i-1;j++)
{ if(X[j]>X[j+1])
{ v=X[j];
X[j]=X[j+1];
X[j+1]=v;
p++;}}
if (p==0)break; }
printf("\n");
for(i=0;i<10;i++)
printf("%d ",X[i]);
printf("\n %d \n",C);
system("PAUSE");
return 0;
}
```

1. 10
2. 8
3. 5
4. 6

9.1.4. Темы лабораторных работ

1. Решение систем линейных алгебраических уравнений методом Гаусса. Численное интегрирование методами центральных прямоугольников, трапеций и Симпсона.
2. Численное решение задачи Коши. Решение нелинейного уравнения

9.2. Методические рекомендации

Учебный материал излагается в форме, предполагающей самостоятельное мышление студентов, самообразование. При этом самостоятельная работа студентов играет решающую роль в ходе всего учебного процесса.

Начать изучение дисциплины необходимо со знакомства с рабочей программой, списком учебно-методического и программного обеспечения. Самостоятельная работа студента включает работу с учебными материалами, выполнение контрольных мероприятий, предусмотренных учебным планом.

В процессе изучения дисциплины для лучшего освоения материала необходимо регулярно обращаться к рекомендуемой литературе и источникам, указанным в учебных материалах; пользоваться через кабинет студента на сайте Университета образовательными ресурсами электронно-библиотечной системы, а также общедоступными интернет-порталами, содержащими научно-популярные и специализированные материалы, посвященные различным аспектам учебной дисциплины.

При самостоятельном изучении тем следуйте рекомендациям:

– чтение или просмотр материала осуществляйте со скоростью, достаточной для индивидуального понимания и освоения материала, выделяя основные идеи; на основании изученного составить тезисы. Освоив материал, попытаться соотнести теорию с примерами из практики;

– если в тексте встречаются незнакомые или малознакомые термины, следует выяснить их значение для понимания дальнейшего материала;

– осмысливайте прочитанное и изученное, отвечайте на предложенные вопросы.

Студенты могут получать индивидуальные консультации, в т.ч. с использованием средств телекоммуникации.

По дисциплине могут проводиться дополнительные занятия, в т.ч. в форме вебинаров. Расписание вебинаров и записи вебинаров публикуются в электронном курсе по дисциплине.

9.3. Требования к оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Согласована на портале № 73817

Для лиц с ограниченными возможностями здоровья и инвалидов предусмотрены дополнительные оценочные материалы, перечень которых указан в таблице 9.4.

Таблица 9.4 – Дополнительные материалы оценивания для лиц с ограниченными возможностями здоровья и инвалидов

Категории обучающихся	Виды дополнительных оценочных материалов	Формы контроля и оценки результатов обучения
С нарушениями слуха	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы	Преимущественно письменная проверка
С нарушениями зрения	Собеседование по вопросам к зачету, опрос по терминам	Преимущественно устная проверка (индивидуально)
С нарушениями опорно-двигательного аппарата	Решение дистанционных тестов, контрольные работы, письменные самостоятельные работы, вопросы к зачету	Преимущественно дистанционными методами
С ограничениями по общемедицинским показаниям	Тесты, письменные самостоятельные работы, вопросы к зачету, контрольные работы, устные ответы	Преимущественно проверка методами, определяющимися исходя из состояния обучающегося на момент проверки

9.4. Методические рекомендации по оценочным материалам для лиц с ограниченными возможностями здоровья и инвалидов

Для лиц с ограниченными возможностями здоровья и инвалидов предусматривается доступная форма предоставления заданий оценочных средств, а именно:

– в печатной форме;
– в печатной форме с увеличенным шрифтом;
– в форме электронного документа;
– методом чтения ассистентом задания вслух;
– предоставление задания с использованием сурдоперевода.

Лицам с ограниченными возможностями здоровья и инвалидам увеличивается время на подготовку ответов на контрольные вопросы. Для таких обучающихся предусматривается доступная форма предоставления ответов на задания, а именно:

– письменно на бумаге;
– набор ответов на компьютере;
– набор ответов с использованием услуг ассистента;
– представление ответов устно.

Процедура оценивания результатов обучения лиц с ограниченными возможностями здоровья и инвалидов по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:
– в форме электронного документа;
– в печатной форме увеличенным шрифтом.

Для лиц с нарушениями слуха:
– в форме электронного документа;
– в печатной форме.

Для лиц с нарушениями опорно-двигательного аппарата:
– в форме электронного документа;
– в печатной форме.

При необходимости для лиц с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения может проводиться в несколько этапов.

Согласована на портале № 73817

ЛИСТ СОГЛАСОВАНИЯ

Рассмотрена и одобрена на заседании кафедры АОИ
протокол № 6 от «10» 12 2020 г.

СОГЛАСОВАНО:

Должность	Инициалы, фамилия	Подпись
Заведующий выпускающей каф. АОИ	А.А. Сидоров	Согласовано, d4895b45-5d88-49f8-b7c7-e8bf0196776a
Заведующий обеспечивающей каф. АОИ	А.А. Сидоров	Согласовано, d4895b45-5d88-49f8-b7c7-e8bf0196776a
Декан ФДО	И.П. Черкашина	Согласовано, 4580bdea-d7a1-4d22-bda1-21376d739cfc

ЭКСПЕРТЫ:

Доцент, каф. АОИ	Ю.В. Морозова	Согласовано, 8461038d-613f-4932-8e22-2b7293a14b92
Доцент, каф. АОИ	Н.Ю. Салмина	Согласовано, ed28a52c-a209-461c-b4ed-4e958affbfc7

РАЗРАБОТАНО:

Доцент, каф. АОИ	Ю.В. Морозова	Разработано, 8461038d-613f-4932-8e22-2b7293a14b92

Согласована на портале № 73817