МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

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

УТВЕРЖДАЮ
Директор департамента образования
_____ П. Е. Троян
«___» _____ 20__ г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Основы алгоритмизации

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: **ФДО, Факультет дистанционного обучения**
Кафедра: **АОИ, Кафедра автоматизации обработки информации**
Курс: **1**
Семестр: **1**
Учебный план набора 2016 года

Распределение рабочего времени

№	Виды учебной деятельности	1 семестр	Всего	Единицы
1	Самостоятельная работа под руководством преподавателя	12	12	часов
2	Лабораторные работы	8	8	часов
3	Контроль самостоятельной работы	2	2	часов
4	Всего контактной работы	22	22	часов
5	Самостоятельная работа	118	118	часов
6	Всего (без экзамена)	140	140	часов
7	Подготовка и сдача зачета	4	4	часов
8	Общая трудоемкость	144	144	часов
			4.0	З.Е.

Контрольные работы: 1 семестр - 1
Зачет: 1 семестр

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шелупанов А.А.
Должность: Ректор
Дата подписания: 28.02.2018
Уникальный программный ключ:
c53e145e-8b20-45aa-9347-a5e4dbb90e8d

Томск 2018

Согласована на портале № 41281

Рассмотрена и одобрена на заседании кафедры
протокол № 318 от « 3 » 7 2018 г.

ЛИСТ СОГЛАСОВАНИЯ

Рабочая программа дисциплины составлена с учетом требований федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки (специальности) 09.03.04 Программная инженерия, утвержденного 12.03.2015 года, рассмотрена и одобрена на заседании кафедры АОИ «___» _____ 20__ года, протокол №_____.

Разработчики:

доцент каф. АОИ	_____	Ю. В. Морозова
ст. преподаватель каф. АОИ	_____	Н. В. Пермякова

Заведующий обеспечивающей каф. АОИ	_____	Ю. П. Ехлаков

Рабочая программа дисциплины согласована с факультетом и выпускающей кафедрой:

Декан ФДО	_____	И. П. Черкашина
Заведующий выпускающей каф. АОИ	_____	Ю. П. Ехлаков

Эксперты:

Доцент кафедры технологий электронного обучения (ТЭО)	_____	Ю. В. Морозова
Доцент кафедры автоматизации обработки информации (АОИ)	_____	Н. Ю. Салмина

Согласована на портале № 41281

1. Цели и задачи дисциплины

1.1. Цели дисциплины

формирование знаний и умений в области методов представления данных в памяти ЭВМ, основных алгоритмов, оперирующих с ними, а также освоение методов решения задач и создание теоретической основы для изучения ряда специальных дисциплин.

1.2. Задачи дисциплины

– развитие навыков алгоритмического мышления.

2. Место дисциплины в структуре ОПОП

Дисциплина «Основы алгоритмизации» (Б1.Б.13) относится к блоку 1 (базовая часть).

Предшествующими дисциплинами, формирующими начальные знания, являются: Информатика и программирование.

Последующими дисциплинами являются: Математическая логика и теория алгоритмов, Объектно-ориентированное программирование.

3. Требования к результатам освоения дисциплины

Процесс изучения дисциплины направлен на формирование следующих компетенций:

– ОПК-1 владением основными концепциями, принципами, теориями и фактами, связанными с информатикой;

В результате изучения дисциплины обучающийся должен:

– **знать** подходы к определению понятия алгоритма; основные свойства алгоритмов; способы представления алгоритмов; основные алгоритмические структуры – следование, ветвление, итерация; структуры данных и типовые алгоритмы их обработки; понимать роль алгоритмизации в современном мире.

– **уметь** строить логически правильные и эффективные программы.

– **владеть** различными способами анализа алгоритмов; современными методами разработки алгоритмов; способами представления алгоритмов.

4. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины составляет 4.0 зачетных единицы и представлена в таблице 4.1.

Таблица 4.1 – Трудоемкость дисциплины

Виды учебной деятельности	Всего часов	Семестры
		1 семестр
Контактная работа (всего)	22	22
Самостоятельная работа под руководством преподавателя (СРП)	12	12
Лабораторные работы	8	8
Контроль самостоятельной работы (КСР)	2	2
Самостоятельная работа (всего)	118	118
Подготовка к контрольным работам	8	8
Оформление отчетов по лабораторным работам	8	8
Подготовка к лабораторным работам	8	8
Самостоятельное изучение тем (вопросов) теоретической части курса	94	94
Всего (без экзамена)	140	140
Подготовка и сдача зачета	4	4
Общая трудоемкость, ч	144	144

Согласована на портале № 41281

Зачетные Единицы	4.0	

5. Содержание дисциплины

5.1. Разделы дисциплины и виды занятий

Разделы дисциплины и виды занятий приведены в таблице 5.1.

Таблица 5.1 – Разделы дисциплины и виды занятий

Названия разделов дисциплины	СРП, ч	Лаб. раб., ч	КСР, ч	Сам. раб., ч	Всего часов (без экзамена)	Формируемые компетенции
1 семестр						
1 Основы алгоритмизации.	2	0	2	20	22	ОПК-1
2 Синтаксис и алфавит языка Си.	4	0		18	22	ОПК-1
3 Конструкции структурного программирования в Си.	2	4		42	48	ОПК-1
4 Алгоритмы на массивах.	4	4		38	46	ОПК-1
Итого за семестр	12	8	2	118	140	
Итого	12	8	2	118	140	

5.2. Содержание разделов дисциплины (самостоятельная работа под руководством преподавателя)

Содержание разделов дисциплин (самостоятельная работа под руководством преподавателя) приведено в таблице 5.2.

Таблица 5.2 – Содержание разделов дисциплин (самостоятельная работа под руководством преподавателя)

Названия разделов	Содержание разделов дисциплины (самостоятельная работа под руководством преподавателя)	Трудоемкость, ч	Формируемые компетенции
1 семестр			
1 Основы алгоритмизации.	Основные понятия и определения. Типы данных. Структурное программирование. Системы кодирования алгоритмов.	2	ОПК-1
	Итого	2	
2 Синтаксис и алфавит языка Си.	Интегрированная среда программирования DEV-CPP. Лексемы языка. Ключевые слова. Идентификаторы. Константы. Литеральные строки. Операторы. Знаки пунктуации. Типы данных языка Си. Сложные типы данных. Подготовка и исполнение программы на языке Си.	4	ОПК-1
	Итого	4	
3 Конструкции структурного программирования в Си.	Следование. Ветвление. Циклы. Примеры использования операторов цикла. Вычисление суммы бесконечного ряда. Вычисления по итерационной формуле. Программирование численных методов.	2	ОПК-1

Согласована на портале № 41281

	Итого	2	
4 Алгоритмы на массивах.	Одномерные массивы. Инициализация массива. Поиск значений в массиве. Сортировка массивов. Многомерные массивы. Инициализация матриц. Печать матриц. Примеры решений задач с использованием матриц. Строки. Инициализация строк. Представление строки в памяти компьютера. Стандартные функции для работы со строками. Примеры решений задач со строками.	4	ОПК-1
	Итого	4	
Итого за семестр		12	

5.3. Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами

Разделы дисциплины и междисциплинарные связи с обеспечивающими (предыдущими) и обеспечиваемыми (последующими) дисциплинами представлены в таблице 5.3.

Таблица 5.3 – Разделы дисциплины и междисциплинарные связи

Наименование дисциплин	№ разделов данной дисциплины, для которых необходимо изучение обеспечивающих и обеспечиваемых дисциплин			
	1	2	3	4
Предшествующие дисциплины				
1 Информатика и программирование	+	+	+	+
Последующие дисциплины				
1 Математическая логика и теория алгоритмов	+			+
2 Объектно-ориентированное программирование	+	+	+	+

5.4. Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий представлено в таблице 5.4.

Таблица 5.4 – Соответствие компетенций, формируемых при изучении дисциплины, и видов занятий

Компетенции	Виды занятий				Формы контроля
	СРП	Лаб. раб.	КСР	Сам. раб.	
ОПК-1	+	+	+	+	Контрольная работа, Проверка контрольных работ, Отчет по лабораторной работе, Зачет, Тест

6. Интерактивные методы и формы организации обучения

Не предусмотрено РУП.

7. Лабораторные работы

Наименование лабораторных работ приведено в таблице 7.1.

Согласована на портале № 41281

Таблица 7.1 – Наименование лабораторных работ

Названия разделов	Наименование лабораторных работ	Трудоемкость, ч	Формируемые компетенции
1 семестр			
3 Конструкции структурного программирования в Си.	Лабораторная работа «Проверка условий»	4	ОПК-1
	Итого	4	
4 Алгоритмы на массивах.	Лабораторная работа «Обработка массивов»	4	ОПК-1
	Итого	4	
Итого за семестр		8	

8. Контроль самостоятельной работы

Виды контроля самостоятельной работы приведены в таблице 8.1.

Таблица 8.1 – Виды контроля самостоятельной работы

№	Вид контроля самостоятельной работы	Трудоемкость (час.)	Формируемые компетенции
1 семестр			
1	Контрольная работа с автоматизированной проверкой	2	ОПК-1
Итого		2	

9. Самостоятельная работа

Виды самостоятельной работы, трудоемкость и формируемые компетенции представлены в таблице 9.1.

Таблица 9.1 – Виды самостоятельной работы, трудоемкость и формируемые компетенции

Названия разделов	Виды самостоятельной работы	Трудоемкость, ч	Формируемые компетенции	Формы контроля
1 семестр				
1 Основы алгоритмизации.	Самостоятельное изучение тем (вопросов) теоретической части курса	18	ОПК-1	Зачет, Контрольная работа, Тест
	Подготовка к контрольным работам	2		
	Итого	20		
2 Синтаксис и алфавит языка Си.	Самостоятельное изучение тем (вопросов) теоретической части курса	16	ОПК-1	Зачет, Контрольная работа, Тест
	Подготовка к контрольным работам	2		
	Итого	18		
3 Конструкции структурного	Самостоятельное изучение тем (вопросов) теоре-	32	ОПК-1	Зачет, Контрольная работа, Отчет по лаборатор-

Согласована на портале № 41281

программирования в Си.	тической части курса			ной работе, Тест
	Подготовка к лабораторным работам	4		
	Оформление отчетов по лабораторным работам	4		
	Подготовка к контрольным работам	2		
	Итого	42		
4 Алгоритмы на массивах.	Самостоятельное изучение тем (вопросов) теоретической части курса	28	ОПК-1	Зачет, Контрольная работа, Отчет по лабораторной работе, Тест
	Подготовка к лабораторным работам	4		
	Оформление отчетов по лабораторным работам	4		
	Подготовка к контрольным работам	2		
	Итого	38		
	Выполнение контрольной работы	2	ОПК-1	Контрольная работа
Итого за семестр		118		
	Подготовка и сдача зачета	4		Зачет
Итого		122		

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

Не предусмотрено РУП.

11. Рейтинговая система для оценки успеваемости обучающихся

Рейтинговая система не используется.

12. Учебно-методическое и информационное обеспечение дисциплины

12.1. Основная литература

1. Пермякова Н. В. Информатика и программирование [Электронный ресурс]: учебное пособие / Н. В. Пермякова. – Томск ФДО, ТУСУР, 2016. – 188 с. Доступ из личного кабинета. - Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 30.07.2018).

12.2. Дополнительная литература

1. Трофимов, В. В. Алгоритмизация и программирование [Электронный ресурс]: учебник для академического бакалавриата / В. В. Трофимов, Т. А. Павловская ; под ред. В. В. Трофимова. — М. Издательство Юрайт, 2018. — 137 с. — (Серия Бакалавр. Академический курс. Модуль.). Доступ из личного кабинета студента. - Режим доступа: https://biblio-online.ru/book/B08DB966-3F96-4B5A-B030-E3CD9085CED4/algoritmizaciya-i-programmirovanie (дата обращения: 30.07.2018).

12.3. Учебно-методические пособия

12.3.1. Обязательные учебно-методические пособия

1. Пермякова Н. В. Информатика и программирование [Электронный ресурс]: методические указания по выполнению лабораторных работ. – Томск ФДО ТУСУР, 2016. – 67 с. Доступ из личного кабинета студента. - Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 30.07.2018).
2. Пермякова Н. В. Информатика и программирование : электронный курс / Н. В. Пермякова. – Томск: ТУСУР, ФДО, 2016. Доступ из личного кабинета студента.

Согласована на портале № 41281

3. Пермякова Н. В. Основы алгоритмизации [Электронный ресурс]: методические указания по организации самостоятельной работы для студентов заочной формы обучения технических направлений, обучающихся с применением дистанционных образовательных технологий / Н. В. Пермякова, Ю. П. Ехлаков. – Томск ФДО, ТУСУР, 2018. Доступ из личного кабинета студента. - Режим доступа: https://study.tusur.ru/study/library/ (дата обращения: 30.07.2018).

12.3.2. 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.

12.4. Профессиональные базы данных и информационные справочные системы

1. При изучении дисциплины рекомендуется использовать базы данных и информационно-справочные системы https://lib.tusur.ru/ru/resursy/bazy-dannyh (со свободным доступом).

13. Материально-техническое обеспечение дисциплины и требуемое программное обеспечение

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

13.1.1. Материально-техническое и программное обеспечение дисциплины

Кабинет для самостоятельной работы студентов

учебная аудитория для проведения занятий лабораторного типа, помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

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

Описание имеющегося оборудования:

- Коммутатор MicroTeak;
- Компьютер PENTIUM D 945 (3 шт.);
- Компьютер GELERON D 331 (2 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:

- 7-zip
- Google Chrome
- Kaspersky Endpoint Security 10 для Windows
- Microsoft Windows
- OpenOffice

13.1.2. Материально-техническое и программное обеспечение для лабораторных работ

Кабинет для самостоятельной работы студентов

учебная аудитория для проведения занятий лабораторного типа, помещение для проведения групповых и индивидуальных консультаций, помещение для проведения текущего контроля и промежуточной аттестации, помещение для самостоятельной работы

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

Согласована на портале № 41281

Описание имеющегося оборудования:
- Коммутатор MicroTeak;
- Компьютер PENTIUM D 945 (3 шт.);
- Компьютер GELERON D 331 (2 шт.);
- Комплект специализированной учебной мебели;
- Рабочее место преподавателя.

Программное обеспечение:
- 7-zip
- DEV C++ (с возможностью удаленного доступа)
- Google Chrome
- Kaspersky Endpoint Security 10 для Windows
- Microsoft Windows
- OpenOffice

13.1.3. 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, 201 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 47, 126 ауд.;
- 634034, Томская область, г. Томск, Вершинина улица, д. 74, 207 ауд.

Состав оборудования:
- учебная мебель;
- компьютеры класса не ниже ПЭВМ INTEL Celeron D336 2.8ГГц. - 5 шт.;
- компьютеры подключены к сети «Интернет» и обеспечивают доступ в электронную информационно-образовательную среду университета.

Перечень программного обеспечения:
- Microsoft Windows;
- OpenOffice;
- Kaspersky Endpoint Security 10 для Windows;
- 7-Zip;
- Google Chrome.

13.2. 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 зрениями** 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.

Согласована на портале № 41281

14. Оценочные материалы и методические рекомендации по организации изучения дисциплины

14.1. Содержание оценочных материалов и методические рекомендации

Для оценки степени сформированности и уровня освоения закрепленных за дисциплиной компетенций используются оценочные материалы в составе:

14.1.1. Тестовые задания

Вопрос №1
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 индекса последнего максимального элемента массива.

Вопрос №2
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 индекса первого максимального элемента массива.

Вопрос №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.

Согласована на портале № 41281

4. Поиск индекса первого максимального элемента массива.

Вопрос №4
Какую задачу решает представленный алгоритм?
алг Вопрос 4 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+1
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.

Вопрос №5
Какую задачу решает представленный алгоритм?
алг Вопрос 5 нач
дано цел X[10]
цел n=10
цел a:=0
для i от 1 до n нц
если не(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.

Вопрос № 6
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. i :=1
6. пока i≤n нц
7. если A[i]>0 то p:=I
8. i:=n
9. i:=i+1
10. кц
11. рез p

1. Алгоритм написан верно.
2. В строке 6.
3. В строке 7.

Согласована на портале № 41281

4. В строке 8.

Вопрос № 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. i :=1
6. пока i≤n нц
7. если A[i]>0 то p:=I
8. i:=p
9. i:=i+1
10. кц
11. рез p

1. Алгоритм написан верно.
2. В строке 7.
3. В строке 8.
4. В строке 11.

Вопрос №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.

Вопрос №9

Чему будет равно значение переменной p после выполнения алгоритма, если элементы массива X ={1, 2, 3, -4, 4, 0, -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
```

Согласована на портале № 41281

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.

Вопрос № 10.
Укажите описание соответствующее фрагменту программы:
float X[10] = {1,3,5,7,9,9,7,5,3,1};
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.

Вопрос № 11.
Укажите описание соответствующее фрагменту программы:
```
Float *X = (float*)malloc(sizeof(float)*10);
int i;
for (i=0;i<10;i++)
X[i]=i+i/3.;
```

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.

Вопрос № 12.
Выберите верные утверждения:
1. Оператор «!=» - оператор «не равно».
2. Оператор «< >» - оператор «не равно».
3. Оператор «%» - оператор целочисленного деления.
4. Оператор «=» - оператор «равно».

Вопрос № 13.
Выберите верные утверждения:
1. Оператор «^» - вычисление побитового XOR.
2. Оператор «*» - оператор прямой адресации.
3. Оператор «< >» - оператор «не равно».
4. Оператор «*» - оператор прямой адресации.

Вопрос №14.
Выберите верные утверждения:
1. В последовательности int z,p; float * x; double m[5]; 17 лексем.
2. В последовательности int m; float b[12]; char* x; 11 лексем.
3. Последовательности float k,i; char z[12]; и float k, I; char z[12]; лексически не эквивалентны.
4. В последовательности int a,b; float * x; 8 лексем.

Вопрос №15
Чему будет равно значение переменной p после выполнения алгоритма, если элементы массива X ={1, -9, 8, 7, 0, 5, -6, 7, 8, 2, 1} ?
```
n=10
цел таб X[n]
для i от 1 до n нц
```

Согласована на портале № 41281

```
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.

Вопрос № 16

Выберите верные утверждения:

1. Алгоритм должен удовлетворять требованию правильности.
2. Алгоритм должен удовлетворять требованию непрерывности действий.
3. Алгоритм должен удовлетворять требованию дискретности действий.
4. Алгоритм должен удовлетворять требованию структурности.

Вопрос № 17

Выберите верные утверждения:

1. Алгоритм должен приводить к правильному по отношению к поставленной задаче решению.
2. Алгоритм должен содержать заданное количество шагов.
3. Алгоритм должен быть непрерывным для всех допустимых исходных данных.
4. Алгоритм должен быть дискретным для всех допустимых исходных данных.

Вопрос №18

Пяти программистам дали задания написать алгоритм суммирования целых чисел A1, A2, …, An. Некоторые программисты допустили ошибки в алгоритме. Какой фрагмент записан верно?

1.
```
цел S:=0
i:=A1
пока I ≤ An нц
S:=S+i
i:=i+1;
кц
рез S
```

2.
```
цел S:=A1
x:=A1
для I от 1 до n нц
S:=S+x
x:=x+1
кц
рез S
```

3.
```
цел S:=0
i:=A1
пока I < An нц
S:=S+i
```

Согласована на портале № 41281

i:=i+1;
кц
рез S

4.
цел S:=0
i:=An
пока I ≥ A1 нц
S:=S+i
i:=i+1;
кц
рез S

Вопрос 19

Пяти программистам дали задания написать алгоритм произведения целых чисел A1, A2, …, An. Некоторые программисты допустили ошибки в алгоритме. Какой фрагмент записан верно?

1.
цел S:=1
i:=A1
пока I ≤ An нц
S:=S*i
i:=i+1;
кц
рез S

2.
цел S:=0
для I от A1 до An нц
S:=S*i
кц
рез S

3.
цел S:=1
x:=An
для i от 1 до n нц
S:=S*x
x:=x+1
кц
рез S

4.
цел S:=0
i:=An
пока I ≥ A1 нц
S:=S*i
i:=i+1;
кц
рез S

Вопрос № 20

Выберите верно записанные идентификаторы:

1. String24
2. 2Summa

Согласована на портале № 41281

3. Point@
4. main

14.1.2. Темы контрольных работ

Основы алгоритмизации

Вопрос № 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

Вопрос № 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

Вопрос № 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;

Вопрос № 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[])
{ 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.

Вопрос №5
Что будет выведено на экран при выполнении следующего фрагмента программы при n =

Согласована на портале № 41281

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.

Вопрос №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.

Вопрос №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; }
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.

Согласована на портале № 41281

2. p - сумма первых 5-ти элементов массива.
3. p - сумма элементов массива с четными индексами.
4. p - сумма элементов массива с нечетными индексами.

Вопрос №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.

Вопрос №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.

Вопрос №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;
printf("%d ",X[i]);}
for(i=0;i<n-1;i++){
p=0;C++;
for(j=0;j<n-i-1;j++)
```

Согласована на портале № 41281

```
{ 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

14.1.3. Зачёт

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

Вопрос №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.

Вопрос №2
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.

Согласована на портале № 41281

Вопрос №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.

Вопрос №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.

Вопрос №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.

Вопрос № 6
Программисту требовалось написать алгоритм поиска индекса первого элемента массива,

Согласована на портале № 41281

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.

Вопрос № 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..

Вопрос №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.

Вопрос №9

Чему будет равно значение переменной p после выполнения алгоритма, если элементы мас-

Согласована на портале № 41281

сива 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.

Вопрос № 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.

Вопрос № 11.
Укажите описание соответствующее фрагменту программы:

```
int *X = (int*)malloc(sizeof(int)*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.

Вопрос № 12.
Выберите верные утверждения:
1. Оператор «->» - непрямой селектор компоненты.
2. Оператор «++» - бинарный оператор.
3. Оператор «%» - оператор целочисленного деления.
4. Оператор «=» - оператор «равно».

Вопрос № 13.
Выберите верные утверждения:
1. Оператор «++» - оператор префиксного (или постфиксного увеличения).
2. Оператор «^» - вычисление степени.
3. Оператор «%» - оператор целочисленного деления.
4. Оператор «*» - оператор прямой адресации.

Согласована на портале № 41281

Вопрос №14.

Выберите верные утверждения:

1. В последовательности int m; float b[12]; char* x; 13 лексем.
2. В последовательности int m; float b[12]; char* x; 11 лексем.
3. Последовательности float k,i; char z[12]; и float k, I; char z[12]; лексически не эквивалентны.
4. В последовательности int a,b; float * x; 7 лексем.

Вопрос №15

Чему будет равно значение переменной p после выполнения алгоритма, если элементы массива X ={10, 9, 8, 7, 6, 5, 6, 7, 8, 2, 1} ?

```
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.

Вопрос № 16

Выберите верные утверждения:

1. Алгоритм должен удовлетворять требованию конечности записи.
2. Алгоритм должен удовлетворять требованию последовательности записи.
3. Алгоритм должен удовлетворять требованию дискретности действий.
4. Алгоритм должен удовлетворять требованию структурности.

Вопрос № 17

Выберите верные утверждения:

1. Алгоритм должен содержать конечное количество шагов.
2. Алгоритм должен содержать заданное количество шагов.
3. Алгоритм должен выполнять заданное количество шагов при решении задачи.
4. Алгоритм должен быть дискретным для всех допустимых исходных данных.

Вопрос №18

Пяти программистам дали задания написать алгоритм суммирования целых чисел A1, A2, …, An. Некоторые программисты допустили ошибки в алгоритме. Какой фрагмент записан верно?

1.

```
цел S:=0
для I от A1 до An нц
S:=S+i
кц
рез S
```

2.

```
цел S:=A1
для I от A1 до An нц
```

Согласована на портале № 41281

S:=S+i
кц
рез S

3.
цел S:=0
i:=A1
пока I < An нц
S:=S+i
i:=i+1;
кц
рез S

4.
цел S:=0
i:=An
пока I ≥ A1 нц
S:=S+i
i:=i+1;
кц
рез S

Вопрос 19

Пяти программистам дали задания написать алгоритм произведения целых чисел A1, A2, …, An. Некоторые программисты допустили ошибки в алгоритме. Какой фрагмент записан верно?

1.
цел S:=1
для I от A1 до An нц
S:=S*i
кц
рез S

2.
цел S:=0
для I от A1 до An нц
S:=S*i
кц
рез S

3.
цел S:=1
i:=A1
пока I < An нц
S:=S*i
i:=i+1;
кц
рез S

4.
цел S:=0
i:=An
пока I ≥ A1 нц
S:=S*i
i:=i+1;

Согласована на портале № 41281

кц
рез S

Вопрос № 20
Выберите верно записанные идентификаторы:
1. sinus
2. 2Summa
3. Sinu*s
4. double

14.1.4. Темы лабораторных работ

Лабораторная работа «Проверка условий»
Лабораторная работа «Обработка массивов»

14.1.1. 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 необходимо осуществлять медленно, выделяя основные идеи; на основании изученного составить тезисы. Освоив материал, попытаться соотнести теорию с примерами из практики;
- если в тексте встречаются термины, следует выяснить их значение для понимания дальнейшего материала;
- необходимо осмысливать прочитанное и изученное, отвечать на предложенные вопросы.

Студенты могут получать индивидуальные консультации с использованием средств телекоммуникации.

По дисциплине могут проводиться дополнительные занятия в форме вебинаров. Расписание вебинаров публикуется в кабинете студента на сайте Университета. Запись вебинара публикуется в электронном курсе по дисциплине.

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

Для лиц с ограниченными возможностями здоровья и инвалидов предусмотрены дополнительные оценочные материалы, перечень которых указан в таблице 14.

Таблица 14 – 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)

Согласована на портале № 41281

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 исходя из состояния обучающегося на момент проверки

14.3. 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 № 41281