



1920

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Филиал федерального государственного бюджетного образовательного
учреждения высшего образования
«Кубанский государственный университет» в г. Геленджике



УТВЕРЖДАЮ

Проректор по работе с филиалами

А.А. Евдокимов

05 2022 г.

Рабочая программа дисциплины

ОП.04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ»

09.02.07 «Информационные системы и программирование»

2022


Рабочая программа ОП.04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» разработана на основе федерального государственного образовательного стандарта среднего профессионального образования (ФГОС СПО) по специальности 09.02.07 «Информационные системы и программирование», утвержденного приказом Министерства образования и науки от 9 декабря 2016 года № 1547 (зарегистрирован Министерством юстиции Российской Федерации 26 декабря 2016г., регистрационный №44936) (далее – ФГОС СПО).

Дисциплина	ОП.04 «Основы алгоритмизации и программирования»	
Форма обучения	очная	
Учебный год	2022-2023	
3 курс		6 семестр
лекции		76 час.
практические занятия		40 час.
лабораторные занятия		36 час.
самостоятельные занятия		2 час.
форма итогового контроля		экзамен

Составитель: преподаватель Галицкая Л. В.

Утверждена на заседании предметной (цикловой) комиссии профессиональных дисциплин программирования в компьютерных системах
Протокол № 10 от 25 мая 2022 г.

Председатель предметной (цикловой) комиссии профессиональных дисциплин специальности **09.02.07 «Информационные системы и программирование»**



Л.А. Благова
подпись

Рецензенты:

Директор ООО «Современные
информационные технологии»



А.В.Сметанин

Системный администратор
ЗАО «Геленджикский дельфинерий»




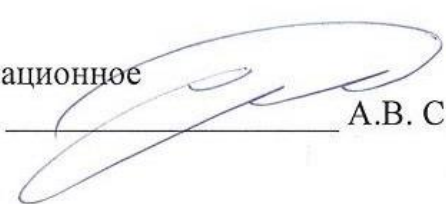
Т.П. Кривошеевко

ЛИСТ
согласования рабочей учебной программы по дисциплине
ОП.04 «Основы алгоритмизации и программирования»
специальность среднего профессионального образования:
09.02.07 Информационные системы и программирование

СОГЛАСОВАНО:

Заместитель директора по УР филиала _____  Т.А. Резуненко

Заведующая сектором библиотеки филиала _____  Л.Г. Соколова

Инженер-электроник (программно-информационное
обеспечение образовательной программы) _____  А.В. Сметанин

СОДЕРЖАНИЕ

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ	5
1.1. Область применения программы.....	5
1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена:	5
1.3. Цели и задачи учебной дисциплины – требования к результатам освоения дисциплины.....	5
1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций).....	6
2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ	12
2.1. Объем учебной дисциплины и виды учебной работы	12
2.2. Структура дисциплины	12
2.3. Тематический план и содержание учебной дисциплины	13
2.4. Содержание разделов дисциплины	15
2.4.1. Занятия лекционного типа	15
2.4.2. Занятия семинарского типа.....	16
2.4.3. Практические занятия и лабораторные работы	16
2.4.4. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	17
3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ.....	18
3.1. Образовательные технологии при проведении лекций	18
3.2. Образовательные технологии при проведении практических занятий и лабораторных работ.....	19
4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ	20
4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине.....	20
4.2. Перечень необходимого программного обеспечения.....	20
5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ.....	21
5.1. Основная литература.....	21
5.2. Дополнительная литература	21
5.3. Периодические издания	22
5.4. Перечень ресурсов, необходимых для освоения дисциплины	22
6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ	23
7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ.....	26
7.1. Паспорт фонда оценочных средств	26
7.2. Критерии оценки знаний.....	26
7.3. Оценочные средства для проведения текущей аттестации.....	27
7.4. Оценочные средства для проведения промежуточной аттестации	28
7.4.1. Примерные вопросы для проведения промежуточной аттестации	28
7.4.2. Практические задания к экзамену	29
8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.....	31

1. ПАСПОРТ РАБОЧЕЙ ПРОГРАММЫ УЧЕБНОЙ ДИСЦИПЛИНЫ ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ

1.1. Область применения программы

Рабочая программа ОП.04 «Основы алгоритмизации и программирования» является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.07. «Информационные системы и программирование»

1.2. Место учебной дисциплины в структуре программы подготовки специалистов среднего звена:

Учебная дисциплина «Основы алгоритмизации и программирования» принадлежит к общепрофессиональному циклу. Она обеспечивает профессиональный уровень подготовки специалиста и соответствует развитию их профессионально значимых качеств.

Для освоения дисциплины обучающиеся используют знания, умения и навыки, сформированные при изучении УДВ.01 «Информатика», ОП.03 Информационные технологии, ОП.10 Численные методы.

1.3. Цели и задачи учебной дисциплины – требования к результатам освоения дисциплины

В результате изучения общепрофессионального цикла обучающийся должен **уметь:**

- разрабатывать алгоритмы для конкретных задач;
- использовать программы для графического отображения алгоритмов;
- определять сложность работы алгоритмов;
- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- оформлять код программы в соответствии со стандартом кодирования;
- выполнять проверку, отладку кода программы.

знать:

- понятие алгоритмизации, свойства алгоритмов, общие принципы построения алгоритмов, основные алгоритмические конструкции;
- эволюцию языков программирования, их классификацию, понятие системы программирования;
- основные элементы языка, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти;
- подпрограммы, составление библиотек подпрограмм;
- объектно-ориентированную модель программирования, основные принципы объектно-ориентированного программирования на примере алгоритмического языка: понятие классов и объектов, их свойств и методов, инкапсуляция и полиморфизма, наследования и переопределения.

Максимальная учебная нагрузка обучающегося 164 часов, в том числе:

- обязательная аудиторная учебная нагрузка обучающегося 152 часов;
- самостоятельная работа обучающегося 2 часа;
- консультация – 4 часа;
- промежуточная аттестация – экзамен.

1.4. Перечень планируемых результатов обучения по дисциплине (перечень формируемых компетенций)

Обучающийся должен обладать **общими** и **профессиональными компетенциями**, включающими в себя способности:

ОК 1. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.

ОК 2. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 4. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 5. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 9. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.

ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием.

ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.

ПК 1.4. Выполнять тестирование программных модулей.

ПК 1.5. Осуществлять рефакторинг и оптимизацию программного кода.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

№ п. п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	практический опыт (владеть)
1.	ОК 1.	Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам.	актуальный профессиональный и социальный контекст, в котором приходится работать и жить; основные источники информации и ресурсы для решения задач и проблем в профессиональном и/или социальном контексте; алгоритмы выполнения работ в профессиональном	распознавать задачу и/или проблему в профессиональном и/или социальном контексте; анализировать задачу и/или проблему и выделять её составные части; определять этапы решения задачи; выявлять и эффективно искать информацию, необходимую для решения задачи и/или проблемы; составить план действия;	

			ной и смежных областях; методы работы в профессиональной и смежных сферах; структуру плана для решения задач; порядок оценки результатов решения задач профессиональной деятельности	определить необходимые ресурсы; владеть актуальными методами работы в профессиональной и смежных сферах; реализовать составленный план; оценивать результат и последствия своих действий (самостоятельно или с помощью наставника)	
2.	ОК 2.	Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности	номенклатура информационных источников, применяемых в профессиональной деятельности; приемы структурирования информации; формат оформления результатов поиска информации	Определять задачи для поиска информации; определять необходимые источники информации; планировать процесс поиска; структурировать получаемую информацию; выделять наиболее значимое в перечне информации; оценивать практическую значимость результатов поиска; оформлять результаты поиска	
3.	ОК 4.	Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами..	психологические основы деятельности коллектива, психологические особенности личности; основы проектной деятельности	Организовывать работу коллектива и команды; взаимодействовать с коллегами, руководством, клиентами в ходе профессиональной деятельности	
4.	ОК 5.	Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.	особенности социального и культурного контекста; правила оформления документов и построения устных сообщений	Грамотно излагать свои мысли и оформлять документы по профессиональной тематике на государственном языке, проявлять толерантность в рабочем коллективе	
5.	ОК 9.	Использовать информационные	современные средства и	Применять средства	

		технологии профессиональной деятельности	в устройства информатизации; порядок их применения и программное обеспечение в профессиональной деятельности	информационных технологий для решения профессиональных задач; использовать современное программное обеспечение	
6.	ОК 10.	Пользоваться профессиональной документацией на государственном и иностранном языках	правила построения простых и сложных предложений на профессиональные темы; основные общеупотребительные глаголы (бытовая и профессиональная лексика); лексический минимум, относящийся к описанию предметов, средств и процессов профессиональной деятельности; особенности произношения; правила чтения текстов профессиональной направленности	Понимать общий смысл и устройства высказываний на известные темы (профессиональные и бытовые), понимать тексты на базовые профессиональные темы; участвовать в диалогах на знакомые общие и профессиональные темы; строить простые высказывания о себе и о своей профессиональной деятельности; кратко обосновывать и объяснить свои действия (текущие и планируемые); писать простые связные сообщения на знакомые или интересующие профессиональные темы	
7.	ПК 1.1.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.	Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Актуальная нормативно-правовая база в области документирования алгоритмов.	Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием. Оформлять документацию на программные средства. Оценка сложности алгоритма.	Разрабатывать алгоритм решения поставленной задачи и реализовывать его средствами автоматизированного проектирования.

8.	ПК 1.2.	Разрабатывать программные модули в соответствии с техническим заданием.	Основные этапы разработки программного обеспечения. Основные принципы технологии структурного и объектно-ориентированного программирования. Знание API современных мобильных операционных систем.	Создавать программу по разработанному алгоритму как отдельный модуль. Оформлять документацию на программные средства. Осуществлять разработку кода программного модуля на языках низкого уровня и высокого уровней в том числе для мобильных платформ.	Разрабатывать код программного продукта на основе готовой спецификации на уровне модуля. Разрабатывать мобильные приложения.
9.	ПК 1.3.	Выполнять отладку программных модулей с использованием специализированных программных средств.	Основные принципы отладки и тестирования программных продуктов. Инструментарий отладки программных продуктов.	Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства. Применять инструментальные средства отладки программного обеспечения.	Использовать инструментальные средства на этапе отладки программного продукта. Проводить тестирование программного модуля по определенному сценарию.
10.	ПК 1.4.	Выполнять тестирование программных модулей.	Основные виды и принципы тестирования программных продуктов.	Выполнять отладку и тестирование программы на уровне модуля. Оформлять документацию на программные средства.	Проводить тестирование программного модуля по определенному сценарию. Использовать инструментальные средства на этапе тестирования программного продукта.
11.	ПК 1.5.	Осуществлять рефакторинг и оптимизацию программного кода.	Способы оптимизации и приемы рефакторинга. Инструментальные средства анализа алгоритма. Методы организации рефакторинга и оптимизации кода. Принципы работы с системой контроля версий.	Выполнять оптимизацию и рефакторинг программного кода. Работать с системой контроля версий.	Анализировать алгоритмы, в том числе с применением инструментальных средств. Осуществлять рефакторинг и оптимизацию программного кода.
12.	ПК 2.4.	Осуществлять разработку тестовых наборов и тестовых	Модели процесса разработки	Использовать выбранную систему контроля	Разрабатывать тестовые наборы (пакеты) для программного модуля.

		<p>сценариев для программного обеспечения.</p>	<p>программного обеспечения. Основные принципы процесса разработки программного обеспечения. Основные подходы к интегрированию программных модулей. Основы верификации и аттестации программного обеспечения. Методы и способы идентификации сбоев и ошибок при интеграции приложений. Методы и схемы обработки исключительных ситуаций. Основные методы и виды тестирования программных продуктов. Приемы работы с инструментальными средствами тестирования и отладки. Стандарты качества программной документации. Основы организации инспектирования и верификации. Встроенные и основные специализированные инструменты анализа качества программных продуктов. Методы организации</p>	<p>версий. Анализировать проектную и техническую документацию. Выполнять тестирование интеграции. Организовывать постобработку данных. Использовать приемы работы в системах контроля версий. Оценивать размер минимального набора тестов. Разрабатывать тестовые пакеты и тестовые сценарии. Выполнять ручное и автоматизированное тестирование программного модуля. Выявлять ошибки в системных компонентах на основе спецификаций.</p>	<p>Разрабатывать тестовые сценарии программного средства. Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.</p>
--	--	--	---	---	---

			работы в команде разработчиков.		
13.	ПК 2.5	Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.	<p>Модели процесса разработки программного обеспечения.</p> <p>Основные принципы процесса разработки программного обеспечения.</p> <p>Основные подходы к интегрированию программных модулей.</p> <p>Основы верификации и аттестации программного обеспечения.</p> <p>Стандарты качества программной документации.</p> <p>Основы организации инспектирования и верификации.</p> <p>Встроенные и основные специализированные инструменты анализа качества программных продуктов.</p> <p>Методы организации работы в команде разработчиков.</p>	<p>Использовать выбранную систему контроля версий.</p> <p>Использовать методы для получения кода с заданной функциональностью и степенью качества.</p> <p>Анализировать проектную и техническую документацию.</p> <p>Организовывать постобработку данных. Приемы работы в системах контроля версий.</p> <p>Выявлять ошибки в системных компонентах на основе спецификаций.</p>	Инспектировать разработанные программные модули на предмет соответствия стандартам кодирования.

2. СТРУКТУРА И СОДЕРЖАНИЕ УЧЕБНОЙ ДИСЦИПЛИНЫ

2.1. Объем учебной дисциплины и виды учебной работы

Вид учебной деятельности	Объем часов
Максимальная учебная нагрузка (всего)	164
Обязательная аудиторная учебная нагрузка (всего)	152
в том числе:	
лекции	76
практические занятия	40
лабораторные занятия	36
Самостоятельная работа обучающегося	2
в том числе:	
<i>Самостоятельная внеаудиторная работа в виде домашних практических заданий, индивидуальных заданий, самостоятельного подбора и изучения дополнительного теоретического материала</i>	2
Консультация	4
Итоговая аттестация (экзамен)	6

2.2. Структура дисциплины

Наименование разделов и тем	Количество аудиторных часов			Самостоятельная работа студента (час)
	Всего	Теоретическое обучение	Практические и лабораторные занятия	
Раздел 1. Введение в программирование	12	4	6	
Тема 1.1. Языки программирования	8	4	4	
Тема 1.2. Типы данных	4	2	2	
Раздел 2.	40	20	20	
Тема 2.1. Операторы языка программирования	40	20	20	
Раздел 3.	30	14	14	2
Тема 3.1. Процедуры и функции	12	6	6	
Тема 3.2. Структуризация в программировании	6	2	2	2
Тема 3.3. Модульное программирование	12	6	6	
Раздел 4. Основные конструкции языков программирования	12	6	6	
Тема 4.1 Указатели	12	6	6	
Раздел 5	60	30	30	
Тема 5.1. Основные принципы объектноориентированного программирования (ООП)	12	6	6	
Тема 5.2. Интегрированная среда разработчика	12	6	6	
Тема 5.3. Визуальное событийноуправляемое программирование	10	4	6	
Тема 5.4. Разработка оконного приложения	12	6	6	
Тема 5.5. Этапы разработки приложений	10	6	4	
Тема 5.6. Иерархия классов	4	2	2	
Всего по дисциплине	154	76	76	2

2.3. Тематический план и содержание учебной дисциплины

Наименование разделов и тем	Содержание учебного материала, лабораторные работы и практические занятия, самостоятельная работа обучающихся, курсовая работа (проект)	Объем часов	Уровень освоения
Раздел 1. Введение в программирование		12	
Тема 1.1. Языки программирования	Содержание учебного материала	4	
	1 Развитие языков программирования. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы	2	2
	2 Жизненный цикл программы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере	2	2
	Практические занятия и лабораторные работы:	4	2,3
	1 Знакомство со средой программирования	4	
Самостоятельная работа обучающихся: Не предусмотрено			
Тема 1.2. Типы данных	Содержание учебного материала	2	
	1 Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных	2	2
	Практические занятия и лабораторные работы:	2	2,3
	1 Составление программ линейной структуры.	2	
	Самостоятельная работа обучающихся: Не предусмотрено		
Раздел 2.		40	
Тема 2.1. Операторы языка программирования	Содержание учебного материала	20	
	1 Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор.	2	2
	2 Условный оператор. Оператор выбора.	2	2
	3 Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы	4	2
	4 Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками.	4	2
	5 Структурированный тип данных – множество. Операции над множествами.	4	2
	6 Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа	4	2
	Практические занятия и лабораторные работы:	20	2,3
	1 Составление программ циклической структуры	10	
2 Обработка массивов	10	2,3	
Раздел 3.		30	
Тема 3.1. Процедуры и функции	Содержание учебного материала	6	
	1 Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций	4	2
	2 Рекурсия. Программирование рекурсивных алгоритмов.	2	2
	Практические занятия и лабораторные работы:	6	2,3
1 Применение рекурсивных функций.	6		
Тема 3.2. Структуризация в программировании	Содержание учебного материала	2	
	1 Основы структурного программирования. Методы структурного программирования	2	2
	Практические занятия и лабораторные работы:	2	2,3
	1 Решение задач	2	
Самостоятельная работа обучающихся:		2	
1 Программирование модуля.	2	2,3	
Тема 3.3.	Содержание учебного материала	6	

Модульное программирование	1	Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы	4	2
	2	Стандартные модули.	2	2
	Практические занятия и лабораторные работы:		6	2,3
	1	Решение задач	6	
Раздел 4. Основные конструкции языков программирования				
Тема 4.1 Указатели	Содержание учебного материала		6	
	1	Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных.	4	2
	2	Структуры данных на основе указателей	2	2
	Практические занятия и лабораторные работы:		6	
	1	Задача о стеке	6	2,3
Раздел 5.			60	
Тема 5.1. Основные принципы объектноориентированного программирования (ООП)	Содержание учебного материала		6	
	1	История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс	2	2
	2	Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Классы объектов. Компоненты и их свойства	2	2
	3	Событийно-управляемая модель программирования. Компонентноориентированный подход	2	2
	Практические занятия и лабораторные работы:		6	
	1	Решение задач	6	2,3
Тема 5.2. Интегрированная среда разработчика	Содержание учебного материала		6	
	1	Требования к аппаратным и программным средствам интегрированной среды разработчика	2	2
	2	Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов	2	2
	3	Панель компонентов и их свойства. Окно кода проекта. Настройка среды и параметров проекта	2	2
	Практические занятия и лабораторные работы:		6	
	1	Создание проекта с использованием компонентов для работы с текстом	6	2,3
Тема 5.3. Визуальное событийноуправляемое программирование	Содержание учебного материала		4	
	1	Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение	2	2
	2	Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий	2	2
	Практические занятия и лабораторные работы:		6	
	1	Создание процедур на основе событий.	6	2,3
Тема 5.4. Разработка оконного приложения	Содержание учебного материала		6	
	1	Разработка функционального интерфейса приложения. Создание интерфейса приложения	2	2
	2	Разработка функциональной схемы работы приложения	2	2
	3	Разработка игрового приложения	2	2
	Практические занятия и лабораторные работы:		6	
1	Разработка оконного приложения с несколькими формами.	6	2,3	
Тема 5.5. Этапы разработки приложений	Содержание учебного материала		6	
	1	Разработка приложения	2	2
	2	Проектирование объектно-ориентированного приложения. Тестирование, отладка приложения	2	2

	3	Создание интерфейса пользователя	2	2
	Практические занятия и лабораторные работы:		4	
	1	Программирование приложений.	4	2,3
Тема 5.6. Иерархия классов	Содержание учебного материала		2	
	1	Классы ООП: виды, назначение, свойства, методы, события	2	2
	Практические занятия и лабораторные работы:		2	
	1	Решение задач	2	2,3
Итого			70	

Для характеристики уровня освоения учебного материала используются следующие обозначения: 1. – ознакомительный (узнавание ранее изученных объектов, свойств); 2. – репродуктивный (выполнение деятельности по образцу, инструкции или под руководством) 3. – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач)

2.4. Содержание разделов дисциплины

2.4.1. Занятия лекционного типа

№	Наименование темы	Содержание темы	Форма текущего контроля
1.1.	Языки программирования	Развитие языков программирования. Обзор языков программирования. Области применения языков программирования. Стандарты языков программирования. Среда проектирования. Компиляторы и интерпретаторы Жизненный цикл программы. Программа. Программный продукт и его характеристики. Основные этапы решения задач на компьютере	Т, У
1.2.	Типы данных	Типы данных. Простые типы данных. Производные типы данных. Структурированные типы данных	Т, У
2.1.	Операторы языка программирования	Операции и выражения. Правила формирования и вычисления выражений. Структура программы. Ввод и вывод данных. Оператор присваивания. Составной оператор. Условный оператор. Оператор выбора. Цикл с постусловием. Цикл с предусловием. Цикл с параметром. Вложенные циклы. Массивы. Двумерные массивы. Строки. Стандартные процедуры и функции для работы со строками. Структурированный тип данных – множество. Операции над множествами. Комбинированный тип данных – запись. Файлы последовательного доступа. Файлы прямого доступа	Т, У
3.1.	Процедуры и функции	Общие сведения о подпрограммах. Определение и вызов подпрограмм. Область видимости и время жизни переменной. Механизм передачи параметров. Организация функций. Рекурсия. Программирование рекурсивных алгоритмов.	Т, У
3.2.	Структуризация в программировании	Основы структурного программирования. Методы структурного программирования	Т, У
3.3.	Модульное программирование	Модульное программирование. Понятие модуля. Структура модуля. Компиляция и компоновка программы. Стандартные модули.	Т, У
4.1.	Указатели	Указатели. Описание указателей. Основные понятия и применение динамически распределяемой памяти. Создание и удаление динамических переменных. Структуры данных на основе указателей.	Т, У

№	Наименование темы	Содержание темы	Форма текущего контроля
5.1.	Основные принципы объектноориентированного программирования (ООП)	История развития ООП. Базовые понятия ООП: объект, его свойства и методы, класс, интерфейс. Основные принципы ООП: инкапсуляция, наследование, полиморфизм. Классы объектов. Компоненты и их свойства. Событийно-управляемая модель программирования. Компонентноориентированный подход.	Т, У
5.2.	Интегрированная среда разработчика	Требования к аппаратным и программным средствам интегрированной среды разработчика. Интерфейс среды разработчика: характеристика, основные окна, инструменты, объекты. Форма и размещение на ней управляющих элементов. Панель компонентов и их свойства. Окно кода проекта. Настройка среды и параметров проекта.	Т, У
5.3.	Визуальное событийноуправляемое программирование	Основные компоненты (элементы управления) интегрированной среды разработки, их состав и назначение. Свойства компонентов. Виды свойств. Синтаксис определения свойств. Назначения свойств и их влияние на результат. Управление объектом через свойства. События компонентов (элементов управления), их сущность и назначение. Создание процедур на основе событий.	Т, У
5.4.	Разработка оконного приложения	Разработка функционального интерфейса приложения. Создание интерфейса приложения. Разработка функциональной схемы работы приложения. Разработка игрового приложения.	Т, У
5.5.	Этапы разработки приложений	Разработка приложения. Проектирование объектно-ориентированного приложения. Тестирование, отладка приложения. Создание интерфейса пользователя.	Т, У
5.6.	Иерархия классов	Классы ООП: виды, назначение, свойства, методы, события.	Т, У
Примечание: Т – тестирование, У – устный опрос			

2.4.2. Занятия семинарского типа

Не предусмотрено

2.4.3. Практические занятия и лабораторные работы

№	Наименование темы	Наименование практических работ	Форма текущего контроля
1.1.	Языки программирования	Знакомство со средой программирования	ПР, Т, У
1.2.	Типы данных	Составление программ линейной структуры.	ПР, Т, У
2.1.	Операторы языка программирования	Составление программ циклической структуры. Обработка массивов	ПР, Т, У
3.1.	Процедуры и функции	Применение рекурсивных функций.	ПР, Т, У
3.2.	Структуризация в программировании	Решение задач	ПР, Т, У
3.3.	Модульное программирование	Решение задач	ПР, Т, У
4.1.	Указатели	Задача о стеке	ПР, Т, У
5.1.	Основные принципы объектноориентированного программирования (ООП)	Решение задач	ПР, Т, У
5.2.	Интегрированная среда разработчика	Создание проекта с использованием компонентов для работы с текстом	ПР, Т, У
5.3.	Визуальное событийноуправляемое программирование	Создание процедур на основе событий.	ПР, Т, У

№	Наименование темы	Наименование практических работ	Форма текущего контроля
5.4.	Разработка оконного приложения	Разработка оконного приложения с несколькими формами.	ПР, Т, У
5.5.	Этапы разработки приложений	Программирование приложений.	ПР, Т, У
5.6.	Иерархия классов	Решение задач	ПР, Т, У
Примечание: ПР- практическая работа, Т – тестирование, У – устный опрос			

2.4.4. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Самостоятельная работа обучающихся является важнейшей формой учебно-познавательного процесса.

Основная цель самостоятельной работы обучающегося при изучении дисциплины – закрепить теоретические знания, полученные в ход лекционных занятий, а также сформировать практические навыки подготовки в области информационных технологий.

Самостоятельная работа обучающегося в процессе освоения дисциплины включает:

- изучение основной и дополнительной литературы по курсу;
- самостоятельное изучение некоторых вопросов (конспектирование);
- работу с электронными учебными ресурсами;
- изучение материалов периодической печати, интернет ресурсов;
- подготовку к тестированию;
- подготовку к практическим занятиям,
- самостоятельное выполнение домашних заданий,
- подготовку реферата (доклада, эссе) по одной из проблем курса.

На самостоятельную работу студентов отводится 2 часа учебного времени.

№	Наименование темы, вида СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы (имеющегося в библиотеке и/или в ЭБС)
1	Программирование модуля.	<p>1. Голицына, О. Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 431 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-570-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/1150328. - Режим доступа: по подписке.</p> <p>2. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: https://znanium.com/catalog/product/1735805. - Режим доступа: по подписке.</p> <p>3. Семакин, И. Г. Основы алгоритмизации и программирования : учебник для студентов учреждений среднего профессионального образования / И. Г. Семакин, А. П. Шестаков. - 5-е изд., стер. - Москва : Академия, 2021. - 301 с. - (Профессиональное образование. ТОП-50). - Библиогр.: с. 298-299. - ISBN 978-5-4468-9989-0. - Текст : непосредственный</p>

Кроме перечисленных источников по темам самостоятельной работы, обучающийся может воспользоваться Электронно-библиотечными системами (ЭБС), профессиональными базами данных, электронными базами периодических изданий,

другими информационными ресурсами, указанными в разделе 5.4 «Перечень ресурсов информационно-коммуникационной сети «Интернет», необходимых для освоения дисциплины», включающий ресурсы, доступ к которым обеспечен по договорам с правообладателями, и образовательные, научные, справочные ресурсы открытого доступа, имеющие статус официальных (федеральные, отраслевые, учреждений, организаций и т.п.), а также поисковыми системами сети Интернет для поиска и работы с необходимой информацией.

Для освоения данной дисциплины и выполнения предусмотренных учебной программой курса заданий по самостоятельной работе студент может использовать следующее учебно-методическое обеспечение:

- электронный учебник по прикладному программированию;
- методические рекомендации преподавателя к лекционному материалу;
- методические рекомендации преподавателя к практическим занятиям;
- методические рекомендации преподавателя к выполнению самостоятельных домашних заданий.

Началом организации любой самостоятельной работы должно быть привитие навыков и умений грамотной работы с учебной и научной литературой. Этот процесс, в первую очередь, связан с нахождением необходимой для успешного овладения учебным материалом литературой. Студент должен уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

Студенты для полноценного освоения учебного курса должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим (лабораторным) занятиям. Желательно, чтобы конспекты лекций записывались в логической последовательности изучения курса и содержались в одной тетради.

3. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

Для реализации компетентностного подхода предусматривается использование в учебном процессе компьютерных активных и интерактивных форм проведения аудиторных и внеаудиторных занятий с целью формирования и развития профессиональных навыков обучающихся.

В процессе преподавания применяются образовательные технологии развития критического мышления. Обязательны компьютерные практические работы по разделам дисциплины.

В учебном процессе наряду с традиционными образовательными технологиями используются электронные учебники, компьютерное обучение, тестирование, учебные видеофильмы, тематические презентации, интерактивные технологии.

3.1. Образовательные технологии при проведении лекций

№	Тема	Виды применяемых образовательных технологий	Кол-во час
1.1.	Языки программирования	Компьютерные технологии обучения, активное обучение, тестирование.	4*
1.2.	Типы данных		2*
2.1.	Операторы языка программирования		20*
3.1.	Процедуры и функции		6*
3.2.	Структуризация в программировании		2*
3.3.	Модульное программирование		6*

4.1.	Указатели	6*
5.1.	Основные принципы объектноориентированного	6*
5.2.	Интегрированная среда разработчика	6*
5.3.	Визуальное событийноуправляемое программирование	4*
5.4.	Разработка оконного приложения	6*
5.5.	Этапы разработки приложений	6*
5.6.	Иерархия классов	2*
Итого по курсу		76
в том числе интерактивное обучение*		76*

3.2.Образовательные технологии при проведении практических занятий и лабораторных работ

№	Тема занятия	Кол. час	Виды применяемых образовательных технологий
1.1.	Языки программирования	4*	Компьютерные технологии обучения. Активное обучение. Дискуссия по теоретическим вопросам. Решение задач индивидуально. Решение задач малыми группами. Разбор решения задач.
1.2.	Типы данных	2*	
2.1.	Операторы языка программирования	20*	
3.1.	Процедуры и функции	6*	
3.2.	Структуризация в программировании	2*	
3.3.	Модульное программирование	6*	
4.1.	Указатели	6*	
5.1.	Основные принципы объектноориентированного программирования (ООП)	6*	
5.2.	Интегрированная среда разработчика	6*	
5.3.	Визуальное событийноуправляемое программирование	6*	
5.4.	Разработка оконного приложения	6*	
5.5.	Этапы разработки приложений	4*	
5.6.	Иерархия классов	2*	
Итого по курсу		76	
в том числе интерактивное обучение*		76*	

4. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ ДИСЦИПЛИНЫ

4.1. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине

Реализация учебной дисциплины осуществляется в специально оборудованном компьютерном классе.

Оборудование учебного кабинета:

- мультимедийный проектор, экран;
- персональный компьютер, динамики;
- выход в Интернет;
- учебная мебель;
- доска учебная.

Учебно-методическое обеспечение:

Рабочая учебная программа

Фонд оценочных средств

Учебники

Дополнительная литература

Методические разработки к занятиям

Дидактический раздаточный материал

Карточки-задания, тестовые задания

Демонстрационные материалы (мультимедийные презентации, видеофильмы)

4.2. Перечень необходимого программного обеспечения

1. Lazarus – визуальная среда программирования (в свободном доступе);
2. PascalABC - визуальная среда программирования (в свободном доступе);
3. PascalABC.NET - визуальная среда программирования (в свободном доступе);
4. WEB-среда разработки PascalABC.NET. URL <http://wde.pascalabc.net/> (в свободном доступе);
5. Среда разработки ПО PortableDelphi. (в свободном доступе);
6. Разработчик инсталляторов InnoSetup. (в свободном доступе);
7. 7-zip архиватор; (лицензия на англ. <http://www.7-zip.org/license.txt>)
8. Adobe Acrobat Reader просмотрщик файлов ; (лицензия - <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>)
9. Adobe Flash Player –графический редактор; (лицензия - <https://get.adobe.com/reader/?loc=ru&promoid=KLXME>)
10. Apache OpenOffice – офисный пакет; (лицензия - <http://www.openoffice.org/license.html>)
11. FreeCommander - проводник; (лицензия - <https://freecommander.com/ru/%d0%bb%d0%b8%d1%86%d0%b5%d0%bd%d0%b7%d0%b8%d1%8f/>)
12. Google Chrome - браузер;(лицензия - https://www.google.ru/chrome/browser/privacy/eula_text.html)
13. LibreOffice – офисный пакет (в свободном доступе);
14. Mozilla Firefox - браузер.(лицензия - <https://www.mozilla.org/en-US/MPL/2.0/>)
15. nanoCAD версия 5.1 локальная (лицензия - серийный номер: NC50B-45103)
16. Программный комплекс "Универсальный тест 4.0.0.1" <http://www.timk.ru/> (в свободном доступе).

5. ПЕРЕЧЕНЬ ОСНОВНОЙ И ДОПОЛНИТЕЛЬНОЙ УЧЕБНОЙ ЛИТЕРАТУРЫ

5.1. Основная литература

1. Голицына, О. Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 431 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-570-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1150328>. – Режим доступа: по подписке.
2. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В.Д. Колдаев ; под ред. проф. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2022. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1735805>. – Режим доступа: по подписке.
3. **Семакин, И. Г. Основы алгоритмизации и программирования** : учебник для студентов учреждений среднего профессионального образования / И. Г. Семакин, А. П. Шестаков. - 5-е изд., стер. - Москва : Академия, 2021. - 301 с. - (Профессиональное образование. ТОП-50). - Библиогр.: с. 298-299. - ISBN 978-5-4468-9989-0 . - Текст : непосредственный.

5.2 Дополнительная литература

1. Гуриков, С. Р. Основы алгоритмизации и программирования на языке Microsoft Visual Basic : учебное пособие / С.Р. Гуриков. — Москва : ИНФРА-М, 2020. — 594 с. — (Среднее профессиональное образование). - ISBN 978-5-16-014442-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/982532>. – Режим доступа: по подписке.
2. Демин, А. Ю. Информатика. Лабораторный практикум : учебное пособие для среднего профессионального образования / А. Ю. Демин, В. А. Дорофеев. — Москва : Издательство Юрайт, 2022. — 133 с. — (Профессиональное образование). — ISBN 978-5-534-07984-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/494500>
3. Казанский, А. А. Объектно-ориентированный анализ и программирование на Visual Basic 2013 : учебник для среднего профессионального образования / А. А. Казанский. — Москва : Издательство Юрайт, 2022. — 290 с. — (Профессиональное образование). — ISBN 978-5-534-03833-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/491340>
4. Макарова, Н.В., Основы программирования : учебник и практикум / Н.В. Макарова, Ю.Н. Нилова, С.Б. Зеленина, Е.В. Лебедева. — Москва : КноРус, 2021. — 451 с. — ISBN 978-5-406-03394-4. — URL:<https://book.ru/book/936582>. — Текст : электронный.
5. Фризен, И. Г. Основы алгоритмизации и программирования (среда PascalABC.NET) : учебное пособие / И.Г. Фризен. — Москва : ФОРУМ : ИНФРА-М, 2020. — 392 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-005-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1047096> – Режим доступа: по подписке.
6. Черпаков, И. В. Основы программирования : учебник и практикум для среднего профессионального образования / И. В. Черпаков. — Москва : Издательство Юрайт, 2022. — 219 с. — (Профессиональное образование). — ISBN 978-5-9916-9984-6. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/491068>

5.3 Периодические издания

- 1 Открытые системы.- URL: <http://biblioclub.ru/index.php?page=journal&jid=436083>
- 2 Информатика в школе .- URL: <http://dlib.eastview.com/browse/publication/18988>
- 3 Программные продукты и системы.- URL: <http://dlib.eastview.com/browse/publication/64086>
- 4 Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
- 5 Системный администратор.- URL: <http://dlib.eastview.com/browse/publication/66751>
- 6 ComputerwordРоссия.- URL: <http://dlib.eastview.com/browse/publication/64081>
- 7 Мир ПК.- URL: <http://dlib.eastview.com/browse/publication/64067>
- 8 Информационно-управляющие системы.- URL:
<http://dlib.eastview.com/browse/publication/71235>
- 9 Журнал сетевых решений LAN.- URL: <http://dlib.eastview.com/browse/publication/64078>
- 10 Информатика и образование.- URL: <http://dlib.eastview.com/browse/publication/18946>
- 11 Windows IT Pro/ Re.- URL: <http://biblioclub.ru/index.php?page=journal&jid=138741>

5.4 Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. ЭБС «Университетская библиотека ONLINE»: сайт. – URL:<http://biblioclub.ru>
2. ЭБС Издательства «Лань»: сайт. – URL:<http://e.lanbook.com>
3. ЭБС «Юрайт»: сайт. –URL:<https://urait.ru/>
4. ЭБС «BOOK.ru»: сайт. – URL: <https://www.book.ru>
5. ЭБС «ZNANIUM.COM»: сайт. – URL: <https://www.znanium.com>
6. Базы данных компании «Ист Вью»: сайт . –URL: <http://dlib.eastview.com>
7. Научная электронная библиотека «eLibrary.ru»: сайт. – URL: <http://elibrary.ru/>
8. Электронная библиотека "Издательского дома "Гребенников". - URL:
<http://www.grebennikon.ru/>
9. Университетская информационная система РОССИЯ (УИС Россия). - URL:
<http://uisrussia.msu.ru/>
10. "Лекториум ТВ" - видеолекции ведущих лекторов России. - URL: <http://www.lektorium.tv/>
11. База учебных планов, учебно-методических комплексов, публикаций и конференций КубГУ. - URL: <http://docspace.kubsu.ru/>

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ

Обучающиеся для полноценного освоения учебного курса должны составлять конспекты как при прослушивании его теоретической (лекционной) части, так и при подготовке к практическим (семинарским) занятиям. Желательно, чтобы конспекты лекций и семинаров записывались в логической последовательности изучения курса и содержались в одной тетради. Это обеспечит более полную подготовку, как к текущим учебным занятиям, так и сессионному контролю знаний.

Самостоятельная работа является важнейшей формой учебно-познавательного процесса. Цель заданий для самостоятельной работы – закрепить и расширить знания, умения, навыки, приобретенные в результате изучения дисциплины; овладеть умением использовать полученные знания в практической работе; получить первичные навыки профессиональной деятельности.

Началом организации любой самостоятельной работы должно быть привитие навыков и умений грамотной работы с учебной и научной литературой. Этот процесс, в первую очередь, связан с нахождением необходимой для успешного овладения учебным материалом литературой. Учащийся должен изучить список нормативно-правовых актов и экономической литературы, рекомендуемый по учебной дисциплине; уметь пользоваться фондами библиотек и справочно-библиографическими изданиями.

Задания для самостоятельной работы выполняются в письменном виде во внеаудиторное время. Работа должна носить творческий характер, при ее оценке преподаватель в первую очередь оценивает обоснованность и оригинальность выводов. В письменной работе по теме задания учащийся должен полно и всесторонне рассмотреть все аспекты темы, четко сформулировать и аргументировать свою позицию по исследуемым вопросам. Выбор конкретного задания для самостоятельной работы проводит преподаватель, ведущий практические занятия в соответствии с перечнем, указанным в планах практических занятий.

Обучение студентов осуществляется по традиционной технологии (лекции, практики) с включением инновационных элементов.

С точки зрения используемых методов лекции подразделяются следующим образом: информационно-объяснительная лекция, повествовательная, лекция-беседа, проблемная лекция и т. д.

Устное изложение учебного материала на лекции должно конспектироваться. Слушать лекцию нужно уметь – поддерживать своё внимание, понять и запомнить услышанное, уловить паузы. В процессе изложения преподавателем лекции студент должен выяснить все непонятные вопросы. Записывать содержание лекции нужно обязательно – записи помогают поддерживать внимание, способствуют пониманию и запоминанию услышанного, приводят знание в систему, служат опорой для перехода к более глубокому самостоятельному изучению предмета.

Методические рекомендации по конспектированию лекций:

- запись должна быть системной, представлять собой сокращённый вариант лекции преподавателя. Необходимо слушать, обдумывать и записывать одновременно;
- запись ведётся очень быстро, чётко, по возможности короткими выражениями;
- не прекращая слушать преподавателя, нужно записывать то, что необходимо усвоить. Нельзя записывать сразу же высказанную мысль преподавателя, следует её понять и после этого кратко записать своими словами или словами преподавателя. Важно, чтобы в ней не был потерян основной смысл сказанного;

- имена, даты, названия, выводы, определения записываются точно;

- следует обратить внимание на оформление записи лекции. Для каждого предмета заводится общая тетрадь. Отличным от остального цвета следует выделять

отдельные мысли и заголовки, сокращать отдельные слова и предложения, использовать условные знаки, буквы латинского и греческого алфавитов, а также некоторые приёмы стенографического сокращения слов.

Практические занятия по дисциплине «Основы алгоритмизации и программирования» проводятся в основном по схеме:

-устный опрос по теории в начале занятия (обсуждение теоретических проблемных вопросов по теме);

-работа в группах по разрешению различных ситуаций по теме занятия;

-решение практических задач индивидуально;

-подведение итогов занятия (или рефлексия);

-индивидуальные задания для подготовки к следующим практическим занятиям.

Цель практического занятия - научить студентов применять теоретические знания при решении практических задач на основе реальных данных.

На практических занятиях преобладают следующие методы:

-вербальные (преобладающим методом должно быть объяснение);

-практические (письменные задания, групповые задания и т. п.).

Важным для студента является умение рационально подбирать необходимую учебную литературу. Основными литературными источниками являются:

– библиотечные фонды филиала КубГУ в г. Геленджике;

– электронная библиотечная система «Университетская библиотека онлайн»;

– электронная библиотечная система Издательства «Лань».

Поиск книг в библиотеке необходимо начинать с изучения предметного каталога и создания списка книг, пособий, методических материалов по теме изучения.

Просмотр книги начинается с титульного листа, следующего после обложки. На нём обычно помещаются все основные данные, характеризующие книгу: название, автор, выходные данные, данные о переиздании и т.д. На обороте титульного листа даётся аннотация, в которой указывается тематика вопросов, освещённых в книге, определяется круг читателей, на который она рассчитана. Большое значение имеет предисловие книги, которое знакомит читателя с личностью автора, историей создания книги, раскрывает содержание.

Прочитав предисловие и получив общее представление о книге, следует обратиться к оглавлению. Оглавление книги знакомит обучаемого с содержанием и логической структурой книги, позволяет выбрать нужный материал для изучения. Год издания книги позволяет судить о новизне материала. В книге могут быть примечания, которые содержат различные дополнительные сведения. Они печатаются вне основного текста и разъясняют отдельные вопросы. Предметные и алфавитные указатели значительно облегчают повторение изложенного в книге материала. В конце книги может располагаться вспомогательный материал. К нему обычно относятся инструкции, приложения, схемы, ситуационные задачи, вопросы для самоконтроля и т.д.

Для лучшего представления и запоминания материала целесообразно вести записи и конспекты различного содержания, а именно:

– пометки, замечания, выделение главного;

– план, тезисы, выписки, цитаты;

– конспект, рабочая запись, реферат, доклад, лекция и т.д.

Читать учебник необходимо вдумчиво, внимательно, не пропуская текста, стараясь понять каждую фразу, одновременно разбирая примеры, схемы, таблицы, рисунки, приведённые в учебнике.

Одним из важнейших средств, способствующих закреплению знаний, является краткая запись прочитанного материала – составление конспекта. Конспект – это краткое связное изложение содержания темы, учебника или его части, без подробностей и второстепенных деталей. По своей структуре и последовательности конспект должен соответствовать плану учебника. Поэтому важно сначала составить план, а потом писать конспект в виде ответа на вопросы плана. Если учебник разделён на небольшие озаглавленные части, то заголовки можно рассматривать как пункты плана, а из текста каждой части следует записать те мысли, которые раскрывают смысл заголовка.

Требования к конспекту:

- краткость, сжатость, целесообразность каждого записываемого слова;
- содержательность записи - записываемые мысли следует формулировать кратко, но без ущерба для смысла. Объём конспекта, как правило, меньше изучаемого текста в 7-15 раз;
- конспект может быть, как простым, так и сложным по структуре – это зависит от содержания книги и цели её изучения.

Методические рекомендации по конспектированию:

- прежде чем начать составлять конспект, нужно ознакомиться с книгой, прочитать её сначала до конца, понять прочитанное;
- на обложке тетради записываются название конспектируемой книги и имя автора, составляется план конспектируемого текста;
- записи лучше делать при прочтении не одного-двух абзацев, а целого параграфа или главы;
- конспектирование ведётся не с целью иметь определённые записи, а для более полного овладения содержанием изучаемого текста, поэтому в записях отмечается и выделяется всё то новое, интересное и нужное, что особенно привлекло внимание;
- после того, как сделана запись содержания параграфа, главы, следует перечитать её, затем снова обращаться к тексту и проверить себя, правильно ли изложено содержание.

Техника конспектирования:

- конспектируя книгу большого объёма, запись следует вести в общей тетради;
- на каждой странице слева оставляют поля шириной 25-30 мм для записи коротких подзаголовков, кратких замечаний, вопросов;
- каждая страница тетради нумеруется;
- для повышения читаемости записи оставляют интервалы между строками, абзацами, новую мысль начинают с «красной» строки;
- при конспектировании широко используют различные сокращения и условные знаки, но не в ущерб смыслу записанного. Рекомендуется применять общеупотребительные сокращения, например: м.б. – может быть; гос. – государственный; д.б. – должно быть и т.д.
- не следует сокращать имена и названия, кроме очень часто повторяющихся;
- в конспекте не должно быть механического переписывания текста без продумывания его содержания и смыслового анализа.

Для написания реферата необходимо выбрать тему, согласовать ее с преподавателем, подобрать несколько источников по теме, выполнить анализ источников по решению проблемы, обосновать свою точку зрения на решение проблемы.

7. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ УСПЕВАЕМОСТИ

7.1. Паспорт фонда оценочных средств

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1.1.	Языки программирования	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
1.2.	Типы данных	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
2.1.	Операторы языка программирования	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
3.1.	Процедуры и функции	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
3.2.	Структуризация в программировании	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
3.3.	Модульное программирование	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
4.1.	Указатели	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
5.1.	Основные принципы объектноориентированного программирования (ООП)	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
5.2.	Интегрированная среда разработчика	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
5.3.	Визуальное событийноуправляемое программирование	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
5.4.	Разработка оконного приложения	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
5.5.	Этапы разработки приложений	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос
5.6.	Иерархия классов	ОК 1, ОК 2, ОК 4, ОК 5, ОК 9, ОК 10, ПК 1.1-1.5, ПК 2.4, ПК 2.5	Тестирование Устный опрос

7.2. Критерии оценки знаний

Контроль и оценка результатов освоения дисциплины осуществляется преподавателем в процессе проведения практических занятий, тестирования, а также выполнения обучающимися индивидуальных самостоятельных заданий.

Устный опрос. Устный ответ – это развернутый рассказ, включающий теоретические материалы и примеры их применения. Удовлетворительная оценка ставится, если студент демонстрирует неглубокие теоретические знания, проявляет слабо сформированные навыки анализа явлений и процессов, недостаточное умение делать аргументированные выводы и приводить примеры, показывает не достаточно свободное владение монологической речью, терминологией, логичностью и последовательностью изложения, делает ошибки, которые может исправить только при коррекции преподавателем.

Реферат. Продукт самостоятельной работы студента, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной научной (учебно-исследовательской) темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на нее. Реферат оценивается по количеству привлеченных источников, глубине анализа проблемы, качестве обоснования авторской позиции, глубине раскрытия темы. Удовлетворительная оценка ставится, если тема освещена лишь частично, допущены фактические ошибки в содержании реферата, или имеются

существенные отступления от требований к реферированию, или неполные ответы на дополнительные вопросы.

Тест. Система стандартизированных заданий, позволяющая автоматизировать процедуру измерения уровня знаний и умений обучающегося. Тест оценивается по количеству правильных ответов (не менее 50%).

Контрольная работа. Письменная проверочная работа, представляющая собой изложение ответов на теоретические вопросы по содержанию учебной дисциплины и решение практических заданий. Работа оценивается удовлетворительно, если выполнено не менее половины работы или допущено в ней не более двух грубых ошибок или не более одной грубой ошибки и одного недочета, или не более двух-трех негрубых ошибок, или одной негрубой ошибки и трех недочетов, или при отсутствии ошибок, но при наличии 4-5 недочетов.

Практическая работа. Практическая работа – это особый вид деятельности обучающегося, что подразумевает выполнения разноплановых заданий, не связанных с обработкой теоретического материала. Во время выполнения студенту необходимо использовать ранее полученные теоретические знания. Положительная оценка ставится, если выполнены все задания практической работы с замечаниями; студент ответил на все контрольные вопросы с замечаниями.

Критерии оценки знаний студентов в целом по дисциплине:

«отлично» - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений;

«хорошо» - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускает в ответе или в решении задач некоторые неточности;

«удовлетворительно» - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации;

«неудовлетворительно» - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

7.3. *Оценочные средств для проведения текущей аттестации*

Текущий контроль проводится в форме:

- индивидуальный устный опрос
- тестирование по теоретическому и практическому материалу
- практическая работа – разработка базы данных
- защита выполненного задания,
- индивидуальный устный опрос.

Форма аттестации	Знания	Умения	Практический опыт (владение)	Личные качества обучающегося	Примеры оценочных средств
Устный (письменный) опрос по темам	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков работы с литературным и источниками	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Контрольные вопросы по темам прилагаются
Практические работы	Контроль знания теоретических основ информатики и информационных технологий, возможностей и принципов использования современной компьютерной техники.	Оценка умения работать с современной компьютерной техникой, использовать возможности вычислительной техники и программного обеспечения при решении практических задач.	Оценка навыков работы с вычислительной техникой, прикладными программным и средствами	Оценка способности оперативно и качественно решать поставленные на практических работах задачи и аргументировать результаты	Темы работ прилагаются
Тестирование	Контроль знаний по определенным проблемам	Оценка умения различать конкретные понятия	Оценка навыков логического анализа и синтеза при сопоставлении конкретных понятий	Оценка способности оперативно и качественно отвечать на поставленные вопросы	Вопросы прилагаются

7.4. Оценочные средств для проведения промежуточной аттестации

Промежуточная аттестация

Форма аттестации	Знания	Умения	Иметь практический опыт	Личные качества студента	Примеры оценочных средств
Экзамен	Контроль знания базовых положений по основам алгоритмизации и программирования	Оценка умения понимать специальную терминологию	Оценка навыков логического анализа задачи придумывать алгоритм.	Оценка способности грамотно и четко излагать материал	Вопросы: прилагаются
		Оценка умения разрабатывать алгоритмы и писать программы на языках высокого уровня.	Оценка навыков переработки алгоритма в компьютерную программу	Оценка способности грамотно и четко излагать ход работы программы и аргументировать результаты	Задачи прилагаются

7.4.1. Примерные вопросы для проведения промежуточной аттестации

1. Приведите примеры алгоритмов из повседневной жизни, сложных, разных видов.
2. Что называется алгоритмом?

3. Какие свойства отличают алгоритмы от остальных конструкций?
4. Какие виды алгоритмов вы знаете?
5. Изобразите графически работу циклического алгоритма.
6. В чем заключается необходимость изучения алгоритмов?
7. Что называется линейным алгоритмом?
8. Приведите примеры линейных алгоритмов.
9. Изобразите графически работу линейного алгоритма.
10. Какие виды алгоритмов вы знаете?
11. Какие свойства отличают алгоритмы от остальных конструкций?
12. В чем заключаются особенности работы линейного алгоритма?
13. Что называется условным алгоритмом?
14. Приведите примеры условных алгоритмов.
15. Изобразите графически работу условного алгоритма.
16. В чем отличия условных алгоритмов от линейных?
17. Что называется циклическим алгоритмом?
18. Приведите примеры циклических алгоритмов.
19. Изобразите графически работу цикла в сочетании с ветвлением.
20. Проведите сравнительный анализ всех видов алгоритмов.

7.4.2. Практические задания к экзамену

Пример итогового теста:

1. Важнейший принцип структурного программирования:
 - любой алгоритм имеет дискретную структуру;
 - алгоритм любой сложности можно построить с помощью следующих базовых структур: линейной, ветвящейся, циклической;
 - современный компьютер — это единство аппаратных средств и программного обеспечения;
 - сущность формализации решаемой задачи заключается в составлении алгоритма;
 - в качестве обязательного этапа создания программы выступает ее тестирование и отладка.
2. Алгоритм — это:
3. Алгоритм называется линейным, если:
 - при его выполнении многократно повторяются одни и те же действия;
 - последовательность выполнения его команд зависит от истинности тех или иных условий;
 - его команды выполняются друг за другом независимо от каких-либо условий;
 - он представим в табличной форме;
 - в нем используются исключительно операторы присваивания.
4. Алгоритм включает в себя ветвление, если:
 - при его выполнении многократно повторяются одни и те же действия;
 - последовательность выполнения его команд зависит от истинности тех или иных условий;
 - его команды выполняются друг за другом независимо от каких-либо условий;
 - он представим в табличной форме;
 - он составлен так, что в каждой программной строке записан только один оператор.
5. Алгоритм называется циклическим, если: при его выполнении многократно
 - повторяются одни и те же действия;

-последовательность выполнения его команд зависит от истинности тех или иных условий;
-его команды выполняются друг за другом независимо от каких-либо условий;
представим в табличной форме;
-он составлен так, что в каждой программной строке записан только один оператор.

6. Что можно назвать алгоритмом?

- схему движения автобусов;
- инструкцию по использованию микроволновой печи;
- расписание уроков;
- схему электрической цепи;
- чертеж дома;
- инструкцию по использованию акварельной краски.

8. ДОПОЛНИТЕЛЬНОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

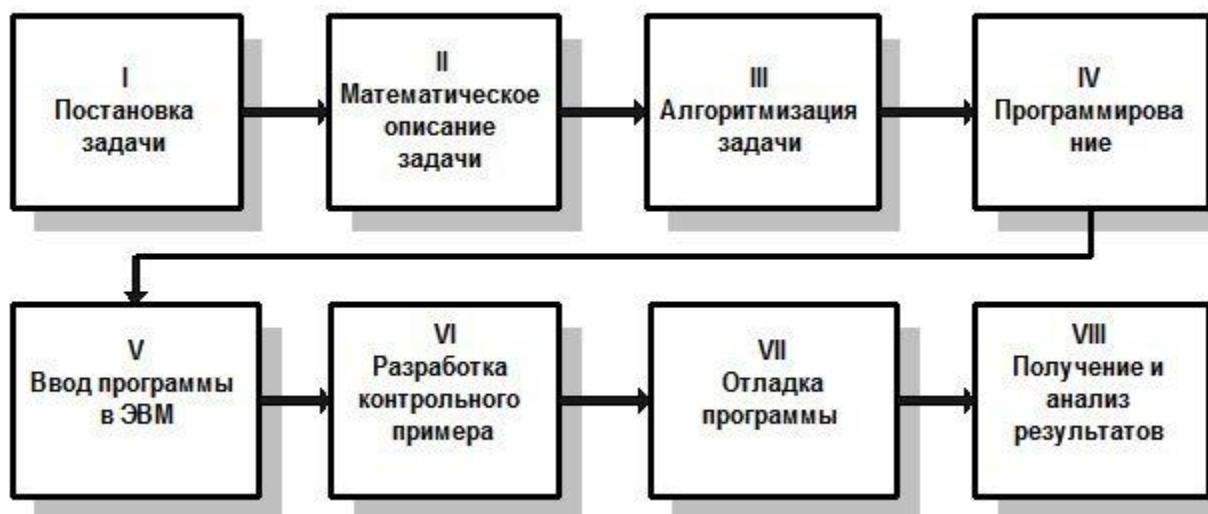
Приложение 1. Краткий конспект лекционных занятий

Задачи:

Формирование базовых знаний по алгоритмизации, рациональные методах разработки алгоритмов;
Изучение базовых алгоритмических структур;
Изучение объектов алгоритмов;
Знакомство с этапами построения математических моделей.

Алгоритмы

Этапы решения задач на ЭВМ



Рассмотрим, из чего складывается процесс решения задачи на ЭВМ, и какие необходимо пройти для этого этапы.

I этап – постановка задачи. На этом этапе должна быть определена предметная область задачи, определены цели задачи, необходимый объем исходной информации, проведено описание каждого исходного данного. Предложен общий подход к решению.

II этап – математическое описание задачи. Цель этого этапа – создать такую математическую модель, которая может быть реализована на компьютере, выбрать оптимальный метод решения.

Под математической моделью будем понимать описание объекта или процесса математическими формулами, связывающими их количественные параметры. При описании математических моделей используются различные системы обозначений.

III этап – алгоритмизация задачи. Главная особенность всех вычислений машины состоит в том, что в основе ее работы лежит программный принцип управления. Это означает, что для решения, как самой простой, так и самой сложной задачи пользователю необходимо использовать перечень инструкций или команд, следуя которым шаг за шагом ЭВМ выдаст необходимый результат.

Таким образом, для того, чтобы решать задачу на ЭВМ, ее необходимо сначала, алгоритмизировать. Именно алгоритмический принцип и лежит в основе работы всех ЭВМ.

На основе математической модели (имеющихся расчетных формул) разрабатывается алгоритм решения. Чаще всего алгоритм разрабатывается на основе блок – схемы с четко определенной последовательностью действий.

IV этап – программирование. Программа – это представление алгоритма с помощью специальных символов, воспринимаемых компьютером. Каждому блоку алгоритма соответствует определенная последовательность операторов. Программа обеспечивает возможность реализации алгоритма и поставленной задачи. При составлении программы возможно уточнение и изменение алгоритма.

V этап – ввод программы в ЭВМ. На этом этапе программу необходимо набрать в интегрированной среде программирования и сохранить на диске.

VI этап – разработка контрольного примера. Для того, чтобы убедиться в правильности составленной программы, необходимо разработать тестовую задачу, проверяющую все ветви алгоритма. Это совокупность таких исходных данных, на основании которых заранее определяются выходные данные.

VII этап – отладка программы. Программа и исходные данные контрольного примера обрабатываются на компьютере и, если контрольный пример работает неправильно, то необходимо найти ошибки, допущенные в программе и вновь проверить ее на контрольном примере.

VIII этап – получение и анализ результатов. После устранения всех ошибок, выявленных тестовой задачей, можно перейти к получению результатов поставленной задачи. Полученные в результате решения результаты необходимо проанализировать.

Алгоритмы

Слово «алгоритм» знакомо многим читателям. Его применяют широко и не только в области вычислительной техники и программирования. В повседневной жизни приходится решать различные задачи. Задачи могут быть бытовые (прибрать комнату, приготовить обед, перейти улицу и т.п.), учебные (решить систему уравнений, построить график) или профессиональные (изготовить деталь на станке, рассчитать заработную плату). Все задачи можно разделить по различным признакам: математические, научные, инженерные, экономические. Одни задачи решаются легко, другие нет. Решить задачу означает получить результат, отвечающий целям данной задачи. Для каждой задачи должно быть известно, что считать результатом. В процессе решения задачи выполняются определенные действия над исходными данными. Совокупность этих действий может быть задана настолько подробно, что ее исполнение становится чисто механическим процессом. Полученная инструкция может использоваться для решения однотипных задач. Такую инструкцию называют алгоритмом.

Понятие алгоритма в информатике является фундаментальным, т.е. таким, которое не определяется через другие, еще более простые понятия.

Содержание понятия алгоритма можно определить следующим образом:

Алгоритм – точное предписание, задающее алгоритмический процесс, начинающийся с произвольного исходного данного и направленный на получение результата, определенного этим исходным данным.

Алгоритм позволяет чисто механически решать любую конкретную задачу из некоторого класса однотипных задач.

Алгоритмический процесс – процесс последовательного преобразования объектов дискретными шагами.

Каждый алгоритм характеризуют независимые параметры:

- совокупность возможных исходных данных (исходные данные могут изменяться в определенных пределах);
- совокупность возможных промежуточных результатов (на каждом шаге должно быть известно, что считать результатом);
- совокупность результатов;
- правило начала;
- правило непосредственной переработки;
- правило окончания;
- правило извлечения результата.

Свойства алгоритма

Понятность. Каждый алгоритм создается для конкретного исполнителя. Чтобы исполнитель мог решить поставленную перед ним задачу, используя алгоритм, он должен уметь выполнить каждое его указание, понимать суть управления. Под «понятностью» алгоритмов понимают указания, понятные исполнителю.

Детерминированность (однозначность). Будучи понятным, алгоритм не должен все же содержать предписаний, смысл которых может восприниматься неоднозначно.

В алгоритмах недопустимы такие ситуации, когда после выполнения очередного предписания алгоритма исполнителю неясно, какое из них должно выполняться на следующем шаге.

Под однозначностью алгоритмов понимается единственность толкования правил выполнения действий и порядка их выполнения.

Дискретность. Под дискретностью понимают возможность разбиения алгоритма на отдельные элементарные действия, выполнение которых человеком или машиной не вызывает сомнения.

Массовость. Очень важно, чтобы составленный алгоритм обеспечивал решение не одной частной задачи, а мог выполнять решение широкого класса задач данного типа.

Под массовостью алгоритмов подразумевается возможность их применения для решения целого класса конкретных задач, отвечающих общей постановке задачи.

Конечность. Выполнение действий, заданных алгоритмом, состоит из конечного числа шагов.

Результативность. Выполнение алгоритма должно завершаться получением определенных результатов. Должно быть известно какой результат должен быть получен через конечное число шагов.

Способы представления алгоритмов

Существует несколько способов представления или записи алгоритмов, отличающихся наглядностью, компактностью, формализацией. Алгоритм может формироваться в виде схемы, текста или программы.

Мы будем рассматривать представление алгоритмов в виде структурных схем – блок-схем, когда отдельные его действия (этапы) изображаются при помощи различных геометрических фигур (блоков), а связи между этапами указываются при помощи стрелок, соединяющих эти фигуры. В блок-схеме отображаются шаги, которые должны выполняться компьютером. Каждому блоку соответствует свой оператор на языке программирования. По составленной блок-схеме пишется программа на языке программирования и реализуется в дальнейшем на компьютере. Существуют также специальные пакеты программ, позволяющие получать результаты непосредственно только по блок-схеме (не используя язык программирования).

Каждый блок имеет свою форму, назначение и определенные размеры. Блоки соединяются соединительными стрелками. Внутри блоков записываются операторы.

Условные графические обозначения блоков

Объекты алгоритмов

У каждой конкретной задачи есть свои реальные объекты (данные, которые подлежат обработке). Каждый объект имеет свои характеристики, свойства или атрибуты. В процессе решения могут возникнуть и вспомогательные объекты. К объектам относятся константы, переменные, файлы, массивы. Каждый объект имеет свой тип (например, число может быть целого типа или вещественного).

Под **константой** будем понимать объект алгоритма, который имеет определенный фиксированный тип и фиксированное, неизменяемое значение.

Под **переменной** будем понимать объект, который имеет определенный фиксированный тип, но значение переменной может меняться на протяжении вычислительного процесса.

Под **массивом** будем понимать упорядоченную совокупность данных одного типа.

Доступ к каждому элементу массива осуществляется по его порядковому номеру (индексу).

Каждый объект в алгоритме должен иметь свое имя – **идентификатор**.

Для обработки объектов алгоритма используют операторы, которые представляют закодированную форму инструкции. Различают простые и составные операторы. К простым относятся операторы присваивания, ввода-вывода, перехода. Группу составных (сложных) операторов представляют операторы условия, циклические, присоединения. Для пояснения алгоритмов используют комментарии.

Базовые алгоритмические конструкции

К базовым конструкциям алгоритмов относятся три основные структуры:

- линейная;
- разветвляющаяся;
- циклическая.

Алгоритм любой сложности состоит из комбинации этих базовых структур.

Линейные алгоритмы

Алгоритм называется линейным, если все действия в нем выполняются последовательно одно за другим.

Типовая блок-схема линейного алгоритма. Линейный алгоритм может содержать блоки ввода и вывода данных, блоки вычислений – действий.

Блоки ввода-вывода могут располагаться в любом месте алгоритма. Все действия данного алгоритма выполняются последовательно одно за другим. К линейным алгоритмам относится большинство расчетных инженерных задач.

Разветвляющиеся алгоритмы

Алгоритм называется разветвляющимся, если выполнение действий в алгоритме происходит после выполнения поставленного условия по одной или другой ветви. Если условие выполняется, то дальнейшее вычисления алгоритма происходят по ветви «Да», если не выполняется, по ветви «Нет». В каждом варианте может быть не одно действие, а несколько. После проверки одного условия, может стоять новая проверка условия, которая тоже будет иметь два выхода.

Циклические алгоритмы

Циклическая структура обеспечивает повторяющуюся реализацию содержащегося в ней функционального узла. Каждый цикл имеет следующие характеристики:

- параметр цикла;
- тело цикла (повторяющиеся действия);
- приращение параметра цикла (изменением его на определенный шаг);
- условие выхода из цикла.

Если заранее известно число повторений в цикле, то такой цикл называют **арифметическим** и его реализуют при помощи типовой схемы «цикл с параметром». Если число повторений в цикле неизвестно, то цикл называют **итерационным**.

Итерационные циклы можно организовывать двумя структурами:

- **с постусловием** ;
- **с предусловием**.

Цикл с постусловием выполняется пока условие «ложно». Этот цикл выполнится всегда хотя бы один раз, так как первая проверка выхода из цикла происходит после выполнения «тела цикла». Цикл с предусловием выполняется пока условие «истинно» и может не выполниться ни одного раза, если при первой проверке условие выхода выполняется. Внутри цикла может содержаться функциональный узел на проверку условия.

Допускается неограниченное соединение базовых структур, их вложение друг в друга, например, цикл с условием, вложенные циклы. Такое соединение позволяет проектировать сложные алгоритмы. Каждой структуре соответствуют свои операторы языка программирования: линейной – операторы присваивания, разветвляющейся – условные операторы, циклической – операторы цикла.

Этапы разработки алгоритма

Разработка любого алгоритма состоит из многих взаимосвязанных этапов. На каждом этапе решаются свои конкретные проблемы, которые в конечном итоге определяют общий результат поставленной задачи.

На первом этапе анализа необходимо понять задачу и определить все составляющие, необходимые для разработки алгоритма. На этом этапе выясняются все исходные данные и требуемые результаты, определяются типы объектов.

Пошаговая детализация задачи позволяет свести трудную задачу к последовательности более простых задач.

Макет исходных данных представляет собой форму исходных данных с записью конкретных значений. В качестве выходного документа служит форма макета вывода или печати результатов, в которой должны быть предусмотрены все поясняющие тексты на выходе.

Таблица идентификаторов может быть представлена в произвольной форме, но она должна содержать все объекты задачи с указанием имен и типов.

Таблица не закрывается и в процессе решения задачи всегда может быть дополнена. При разработке программы таблица используется для записи раздела описаний данных.

Пример 1. Рассмотрим задачу вычисления площади треугольника по трем заданным сторонам.

Постановка задачи. Даны стороны треугольника. Вычислить его площадь.

Математическая модель.

Входные данные: стороны треугольника.

Выходные данные: площадь треугольника.

Дополнительные параметры: полупериметр.

Расчетные формулы:

Данный алгоритм повторяет линейную структуру. Но всегда ли такой алгоритм будет работать? Нет, так как при вычислении площади мы имеем дело с вычислением квадратного корня, подкоренное выражение которого может оказаться отрицательным. Это говорит о том, что не из любых трех отрезков можно построить треугольник. На уточняющем этапе разработки алгоритма необходимо учесть случай существования треугольника и в алгоритме поставить проверку на существование треугольника. Треугольник будет существовать, если сумма двух сторон больше третьей стороны. При такой постановке задачи алгоритм уже будет реализовывать разветвляющуюся структуру. Для исключения варианта некорректного ввода данных, в случае невыполнения условия, сделаем переход на блок ввода данных.

Пример 2. Определить, принадлежит ли точка с координатами (x, y) окружности с центром в начале координат и радиусом r .

Математическая модель.

Входные данные: радиус окружности;
координаты точки.

Выходные данные: сообщение о принадлежности точки окружности.

Дополнительные параметры: расстояние от точки до центра окружности.

Для определения расчетных формул составим геометрическую модель задачи

Точка будет принадлежать окружности при условии, что расстояние от нее до центра окружности будет меньше или равно радиусу окружности.

Расчетные формулы:

Расстояние от точки до центра окружности

Составим таблицу идентификаторов

Алгоритм задачи. Структура алгоритма повторяет типовую разветвляющуюся структуру.

Пример 3. Теперь усложним задачу. Пусть задано n точек. Определить количество точек, принадлежащих окружности. Для решения этой задачи требуется ввести новые объекты. Занесем их в таблицу идентификаторов:

Составим алгоритмы двумя способами: в первом случае используем цикл с параметром, а во втором цикл с постусловием. Циклический блок объединяет следующие действия:

- задание начального значения параметра цикла (переменной i присваивается начальное значение равно 1);
- приращение переменной на 1 ($i=i+1$);
- проверка условия выхода из цикла (как только i достигает значения n , цикл завершается.)

Во втором алгоритме все эти действия показаны в отдельных блоках.

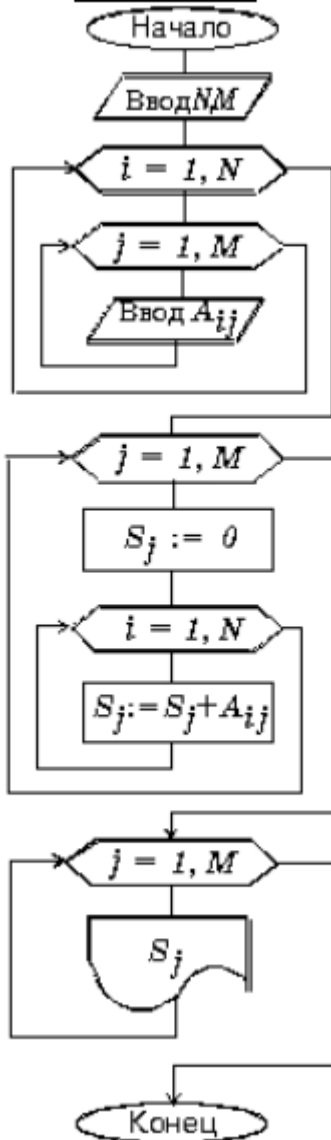
В цикле осуществляется ввод координат точек и подсчет количества точек, принадлежащих окружности.

Приложение 2. Практическая работа по теме «Операторы языка программирования»

ПРИМЕР 1.

Вычислить суммы элементов столбцов заданной матрицы $A(N, M)$.

Блок-схема



Данные	
N=2	$\begin{pmatrix} 2 & 1 \\ 4 & 3 \end{pmatrix}$
M=2	

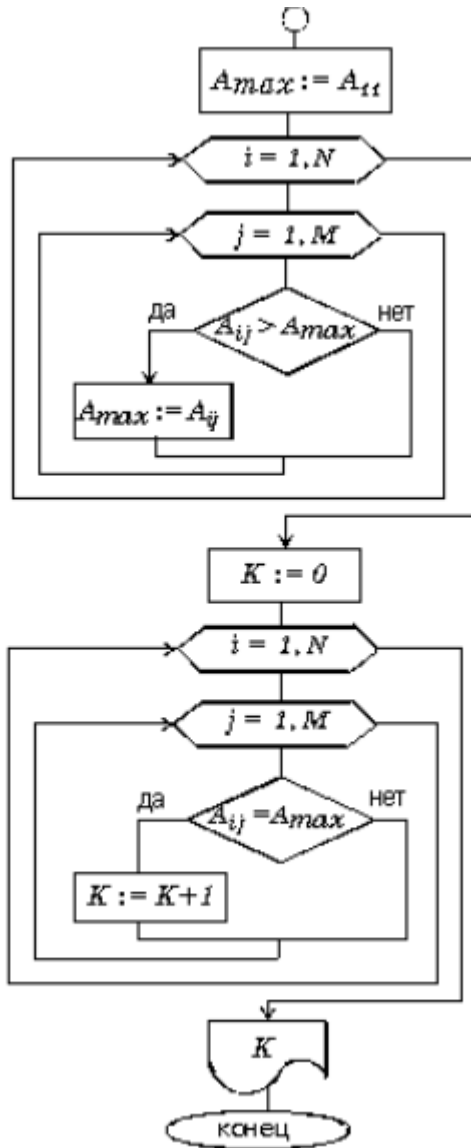
Исполнение алгоритма

j	i	S[i]
1	1	S ₁ =0
	2	S ₁ =0+2=2 S ₁ =2+4=6
2	1	S ₂ =0
	2	S ₂ =0+1=1 S ₂ =1+3=4

ПРИМЕР 2.

Подсчитать, сколько раз встречается в заданной целочисленной матрице $A(N, M)$ максимальное по величине число.

Блок-схема (фрагмент)



Дан	
N=2	
M=3	

Исполнение алгоритма

i	j	$A[i, j] > A_{max}$	A_{max}
1	1	-	1
	2	+	2
	3	+	5
2	1	-	
	2	-	
	3	-	

(продолжение)

i	j	$A[i, j] = A_{max}$	K
1	1	-	0
	2	-	
	3	+	1
2	1	+	2
	2	-	
	3	+	3

Приложение 3.

Экзаменационный билет

Министерство науки и высшего образования Российской Федерации
филиал федерального государственного бюджетного образовательного
учреждения высшего образования
«Кубанский государственный университет» в г. Геленджике
Специальность 09.02.07 «Информационные системы и программирование»
Дисциплина «Основы алгоритмизации и программирования»

Билет №10

1. Алгоритмы вычисления интегралов. Правила написания процедур, параметры входные и выходные.
2. Составить блок-схему алгоритма для выполнения задачи:

В массиве $A(N, N)$ вычислить две суммы элементов, расположенных ниже и выше главной диагонали.

Председатель предметной (цикловой) комиссии Л.А. Благова


ЛИСТ
изменений рабочей учебной программы по дисциплине
ОП.04 ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ


Дополнения и изменения, вносимые в рабочую программу дисциплины

Основания внесения дополнений и изменений	Раздел РПД, в который вносятся изменения	Содержание вносимых дополнений, изменений
Предложение работодателя		
Предложение составителя программы		
Приобретение, издание литературы, обновление перечня и содержания ЭБС, баз данных	Разделы №2.4.5 и №5 Перечень основной и дополнительной учебной литературы	Обновления перечня литературы

Утверждена на заседании предметной (цикловой) комиссии математических и естественно-научных дисциплин протокол № 10 от «25» мая 2022 г.

Заместитель директора по УР филиала _____  Т.А. Резуненко

Заведующая сектором библиотеки филиала _____  Л.Г. Соколова

Инженер-электроник (программно-информационное обеспечение образовательной программы) _____  А.В. Сметанин

РЕЦЕНЗИЯ

на рабочую программу по учебной дисциплине ОП. 04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» для специальности 09.02.07 Информационные системы и программирование

Структура рабочей программы соответствует существующему уровню развития технологий алгоритмизации программирования, она включает в себя описание алгоритмических средств, современного программного обеспечения и соответствующих технических средств.

Рецензируемая программа является частью программы подготовки специалистов среднего звена в соответствии с ФГОС СПО по специальности 09.02.07 «Информационные системы и программирование».

Программа содержит: основные понятия и принципы алгоритмизации задач, конструкции и стандартные приёмы алгоритмизации, основы объектно-ориентированного и визуального программирования, необходимый инструментарий и технологии. Программа имеет достаточную степень полноты и законченности для изучения предмета в условиях СПО.

В программе отражены основные теоретические и практические свойства алгоритмов, что даёт возможность получить необходимые знания о содержании и сущности алгоритмизации, декомпозиции задач, о современном состоянии и тенденциях развития программного обеспечения, компьютерной техники.

Рабочая программа составлена логично, структура дисциплины соответствует принципу единства теоретического и практического обучения, разделы выделены дидактически целесообразно. Последовательность тем, предлагаемых к изучению, направлены на качественное усвоение учебного материала. Виды самостоятельных работ позволяют обобщить и углубить изучаемый материал, и направлены на закрепление умения разрабатывать алгоритмы типовых задач.

Тематический план и содержание учебной дисциплины раскрывает последовательность прохождения тем, соответствует учебному плану и распределению часов. В программе реализованы дидактические принципы обучения: целостность, структурность; отражена взаимосвязь между отдельными элементами структуры. Изучение данной дисциплины способствует эффективной и качественной подготовке молодых специалистов в области информационных технологий. Программа учебной дисциплины продумана и ориентирована на подготовку обучающихся к использованию полученных навыков в своей профессиональной деятельности.

Перечень рекомендуемой основной и дополнительной литературы включает общедоступные источники, изданные не позднее 5 лет. Перечисленные Интернет ресурсы актуальны и достоверны.

Таким образом, рабочая программа содержит все необходимые элементы рекомендуемой структуры, обладает достаточной полнотой и законченностью, является ценным практическим документом при преподавании дисциплины «Основы алгоритмизации и программирования» и может быть рекомендована для использования в учебном процессе при подготовке по специальности 09.02.07 Информационные системы и программирование.

Рецензент:

Директор ООО «Современные
информационные технологии»



А.В.Сметанин

РЕЦЕНЗИЯ

на рабочую программу по учебной дисциплине ОП. 04 «ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ» для специальности 09.02.07 Информационные системы и программирование

Рабочая программа учебной дисциплины ОП. 04 Основы алгоритмизации и программирования разработана на основе ФГОС по специальности СПО 09.02.07 «Информационные системы и программирование», утвержденного приказом Министерства образования и науки РФ от 09.12.2016 №1547 (зарегистрирован в Минюсте России 26.12.2016 № 44936).

Учебная дисциплина изучается с целью получения студентами представления об алгоритмизации и о содержании программирования как научных дисциплин, ознакомление их с основными понятиями, принципами, методологией, методиками анализа алгоритмов и программирования. Программа содержит: основные понятия и принципы алгоритмизации задач, конструкции и стандартные приёмы алгоритмизации, основы объектно-ориентированного и визуального программирования, необходимый инструментарий и технологии. Программа имеет достаточную степень полноты и законченности для изучения предмета в условиях СПО.

В рецензируемой программе отражены основные теоретические и практические свойства алгоритмов, что даёт возможность получить необходимые знания о содержании и сущности алгоритмизации, декомпозиции задач, о современном состоянии и тенденциях развития программного обеспечения.

Изучение данной дисциплины способствует эффективной и качественной подготовке молодых специалистов в области информационных технологий. Программа учебной дисциплины продумана и ориентирована на подготовку обучающихся к использованию полученных навыков в своей профессиональной деятельности.

По структуре программа соответствует современным требованиям. Содержание проработано с достаточной степенью подробности и законченности. Перечень рекомендуемой основной и дополнительной литературы включает общедоступные источники, необходимые для освоения (не старше 5 лет).

Программа учебной дисциплины достаточно продумана и ориентирована на подготовку обучающихся к использованию полученных навыков алгоритмизации в своей профессиональной деятельности.

Следовательно, рабочая программа содержит все необходимые элементы рекомендуемой структуры, обладает достаточной полнотой и законченностью, является ценным практическим документом при преподавании дисциплины «Основы алгоритмизации и программирования» и может быть рекомендована для использования в учебном процессе при подготовке по специальности 09.02.07 Информационные системы и программирование.

Рецензент:

Системный администратор
ЗАО «Геленджикский дельфинариум»



Т.П. Кривошеенко