

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
"Казанский (Приволжский) федеральный университет"
Набережночелнинский институт (филиал)
Отделение информационных технологий и энергетических систем



УТВЕРЖДАЮ

Заместитель директора
по образовательной деятельности
НЧИ КФУ

Ахметов Н.Д.

"16" июня 2021 г.

Программа дисциплины
Программирование и алгоритмизация

Направление подготовки: 01.03.02 – Прикладная математика и информатика

Профиль подготовки: отсутствует

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2021

Содержание

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО
2. Место дисциплины (модуля) в структуре ОПОП ВО
3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся
4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий
 - 4.1. Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)
 - 4.2. Содержание дисциплины (модуля)
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
6. Фонд оценочных средств по дисциплине (модулю)
7. Перечень литературы, необходимой для освоения дисциплины (модуля)
8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)
9. Методические указания для обучающихся по освоению дисциплины (модуля)
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)
11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)
12. Средства адаптации преподавания дисциплины (модуля) к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья
13. Приложение №1. Фонд оценочных средств
14. Приложение №2. Перечень литературы, необходимой для освоения дисциплины (модуля)
15. Приложение №3. Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Программу дисциплины разработали старший преподаватель, б/с Грудцына Л.Ю. (Кафедра системного анализа и информатики, Отделение информационных технологий и энергетических систем, Набережночелнинский институт (филиал) КФУ), LJGrudcyna@kpfu.ru

1. Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения ОПОП ВО

Обучающийся, освоивший дисциплину (модуль), должен обладать следующими компетенциями:

Шифр компетенции	Расшифровка приобретаемой компетенции
ОПК-5	Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения
ПК-7	Способен к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения
ПК-8	Способен к разработке стратегии тестирования и управлению процессом тестирования

Обучающийся, освоивший дисциплину (модуль):

Должен знать:

- основные термины и понятия дисциплины программирование и алгоритмизация;
- существующие подходы к проектированию и методы проектирования алгоритмов и программ в области системного и прикладного программного обеспечения;
- понятие интегрированной среды программирования, назначение ее компонентов, последовательность этапов создания приложения, включая процесс тестирования, с использованием языков программирования;

Должен уметь:

- применять на практике базовые принципы построения алгоритмов, включая принцип последовательной детализации для решения задач профессиональной деятельности;
- создавать простые программы на языке программирования C/C++ в среде VisualStudio .NET или аналогичной в рамках идеологий структурного и процедурного программирования;
- организовывать консольный ввод-вывод в программах на языке C/C++, разрабатывать стратегии тестирования программы, управлять процессом тестирования;

Должен владеть:

- навыками разработки и грамотного представления алгоритмов на основе всех основных видов базовых управляющих структур, с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;
- навыками разработки простых консольных алгоритмов в рамках концепций структурного, процедурного программирования;
- навыками самостоятельной работы с литературой, иными источниками информации по дисциплине.

Должен демонстрировать способность и готовность:

- применять результаты освоения дисциплины в профессиональной деятельности.

2. Место дисциплины (модуля) в структуре ОПОП ВО

Данная дисциплина (модуль) включена в блок "Дисциплины (модули)" основной профессиональной образовательной программы 01.03.02 "Прикладная математика и информатика" и относится к обязательным дисциплинам.

Осваивается на 1 курсе в 1, 2 семестрах.

3. Объем дисциплины (модуля) в зачетных единицах с указанием количества часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 7 зачетных единиц на 252 часа.

Контактная работа - 72 часа, в том числе лекции - 36 часов, практические занятия - 0 часов, лабораторные работы - 36 часов, контроль самостоятельной работы - 0 часов.

Самостоятельная работа - 108 часов.

Контроль (зачёт / экзамен) - 72 часа.

Форма промежуточного контроля дисциплины: экзамен в 1 семестре; экзамен во 2 семестре.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1 Структура и тематический план контактной и самостоятельной работы по дисциплине (модулю)

N	Разделы дисциплины / модуля	Семестр	Виды и часы контактной работы, их трудоемкость (в часах)			Самостоятельная работа
			Лекции	Практические занятия	Лабораторные работы	
1.	Тема 1. Информационно-логические основы работы ЭВМ	1	2	0	0	6
2.	Тема 2. Алгоритмы и алгоритмизация	1	2	0	0	6
3.	Тема 3. Проектирование алгоритмов и программ	1	2	0	2	10
4.	Тема 4. Лексика языка C++. Структура программы	1	6	0	6	25
5.	Тема 5. Массивы, символьные строки и структуры	1	6	0	10	25
6.	Тема 6. Функции	2	8	0	10	16
7.	Тема 7. Файловый ввод-вывод	2	4	0	4	10
8.	Тема 8. Указатели и динамическое распределение памяти. Ссылки	2	6	0	4	10
	Итого		36	0	36	108

4.2 Содержание дисциплины (модуля)

Тема 1. Информационно-логические основы работы ЭВМ

Устройство персонального компьютера, назначение микропроцессора, оперативной памяти; устройства ввода-вывода; логическая структура компьютера; представление числовой и символьной информации в ЭВМ; виды и назначение систем счисления. Системы счисления и формы представления чисел. Варианты представления информации в ПК.

Тема 2. Алгоритмы и алгоритмизация

Понятия алгоритма, исполнителя алгоритма, системы команд исполнителя; свойства алгоритма с иллюстрацией их на примере алгоритма Евклида (или иного); способы отображения алгоритмов (блок-схемы, графы, псевдокод); этапы подготовки и решения задач на компьютере; понятия исходных и выходных данных; способы ввода и вывода данных.

Тема 3. Проектирование алгоритмов и программ

Три основных подхода к проектированию алгоритмов и программ; структурное проектирование алгоритмов, базовые управляющие структуры (следование, ветвление, повторение); средства для создания приложений; понятие языка программирования, классификация языков программирования; средства для создания приложений.

Тема 4. Лексика языка C++. Структура программы

Краткая история языка. Лексика языка C++. Идентификаторы и типы данных. Переменные и константы. Объявление переменных и констант. Инструкции консольного ввода и вывода. Инструкция присваивания. Структура программы и процесс создания программы. Функции и данные. Главная функция. Выражения. Инструкция присваивания. Реализация базовых управляющих структур в языке C/C++.

Тема 5. Массивы, символьные строки и структуры

Понятие о массивах. Особенности массивов. Использование массивов в C++. Виды массивов и их объявление. Обращение к данным массивов. Проблема использования многомерных массивов. Понятие символьных и строковых данных; C-строки и действия над ними; понятие структуры их назначения. Основные стандартные строковые функции.

Тема 6. Функции

Подпрограмма как основной элемент программы в процедурном программировании; назначение подпрограмм, преимущества их использования в программировании; определение и вызов функций в C/C++; формальные и фактические аргументы функций; способы возврата результата из функций; передача массивов в функции через формальные аргументы.

Тема 7. Файловый ввод-вывод

Понятия файла, файлового ввода-вывода; бинарные и текстовые файлы; файловая переменная; основные этапы реализации файлового ввода-вывода. Основные функции для файлового ввода-вывода в стиле языка C/C++: открытие/закрытие, ввод и вывод в бинарных файлах (чтение и запись), ввод и вывод данных из/в текстовых файлах (посимвольный, построчный).

Тема 8. Указатели и динамическое распределение памяти. Ссылки

Понятие указателя; логическая структура оперативной памяти, адресное пространство; принцип динамического выделения памяти; тип ссылки, его отличие от типа указателя. Области применения указателей и ссылок. Проблемы, возникающие при использовании указателей: утечка памяти, неинициализированные указатели. Связь массивов и указателей.

5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа

подразделяется на самостоятельную работу на аудиторных занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования - программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства образования и науки Российской Федерации от 5 апреля 2017 года №301)

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений"

Устав федерального государственного автономного образовательного учреждения "Казанский (Приволжский) федеральный университет"

Правила внутреннего распорядка федерального государственного автономного образовательного учреждения высшего профессионального образования "Казанский (Приволжский) федеральный университет"

Локальные нормативные акты Казанского (Приволжского) федерального университета

6. Фонд оценочных средств по дисциплине (модулю)

Фонд оценочных средств по дисциплине (модулю) включает оценочные материалы, направленные на проверку освоения компетенций, в том числе знаний, умений и навыков. Фонд оценочных средств включает оценочные средства текущего контроля и оценочные средства промежуточной аттестации.

В фонде оценочных средств содержится следующая информация:

- соответствие компетенций планируемым результатам обучения по дисциплине (модулю);
- индикаторы оценивания сформированности компетенций;
- механизм формирования оценки по дисциплине (модулю);
- описание порядка применения и процедуры оценивания для каждого оценочного средства;
- критерии оценивания для каждого оценочного средства;
- содержание оценочных средств, включая требования, предъявляемые к действиям обучающихся, демонстрируемым результатам, задания различных типов.

Фонд оценочных средств по дисциплине находится в Приложении 1 к программе дисциплины (модулю).

7. Перечень литературы, необходимой для освоения дисциплины (модуля)

Освоение дисциплины (модуля) предполагает изучение учебной литературы. Литература может быть доступна обучающимся в одном из двух вариантов (либо в обоих из них):

- в электронном виде - через электронные библиотечные системы на основании заключенных КФУ договоров с правообладателями и предоставленных доступов НЧИ КФУ;
- в печатном виде - в фонде библиотеки Набережночелнинского института (филиала) КФУ. Обучающиеся получают учебную литературу на абонементе по читательским билетам в соответствии с правилами пользования библиотекой.

Электронные издания доступны дистанционно из любой точки при введении обучающимся своего логина и пароля от личного кабинета в системе "Электронный университет". При использовании печатных изданий библиотечный фонд должен быть укомплектован ими из расчета не менее 0,25 экземпляра каждого из изданий, указанных в рабочей программе дисциплины, на одного обучающегося из числа лиц, одновременно осваивающих данную дисциплину.

Перечень учебной литературы, необходимой для освоения дисциплины (модуля), находится в Приложении 2 к рабочей программе дисциплины. Он подлежит обновлению при изменении условий договоров КФУ с правообладателями электронных изданий и при изменении комплектования фондов библиотеки Набережночелнинского института (филиала) КФУ.

8. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

RSDN : сайт, посвященный разработке программного обеспечения - <http://rsdn.ru/>

Клуб программистов - <http://www.programmersclub.ru/>

Основы программирования : электронный курс - <http://www.intuit.ru/studies/courses/2193/67/info>

9. Методические указания для обучающихся по освоению дисциплины (модуля)

Вид работ	Методические рекомендации
лекции	Во время лекционных занятий студенту рекомендуется вести краткий конспект, фиксируя основные теоретические положения изучаемых разделов дисциплины. При подготовке к устному опросу, контрольным работам и экзаменам следует в первую очередь обращаться к конспекту лекций по дисциплине. При этом работа с конспектом лекций и другими литературными источниками должна проводиться систематически, в процессе этой работы студент должен стараться получить полное представление об интересующих его вопросах, особенно, если возникли трудности в понимании какой-то темы. В случае применения в образовательном процессе дистанционных образовательных технологий обучающиеся работают на следующих платформах и ресурсах: в команде "Microsoft Teams". Лекции проводятся в режиме видеособрания в соответствии с расписанием.
лабораторные работы	Лабораторные занятия проводятся в компьютерном классе. Задания по лабораторным работам, рекомендации по их выполнению и примеры выполнения находятся в соответствующих методических материалах. В качестве общих рекомендаций по выполнению лабораторных заданий следует отметить следующие. Перед началом выполнения задания желательно проработать соответствующий теоретический материал, внимательно изучить иллюстрирующие методические примеры. Далее студент должен детально разобрать пример решения схожего выполняемого задания (если есть), внимательно проанализировать программный код. В результате студент должен сформировать максимально четкое и конкретное представление о том, как выполняется программа в примере, какой смысл и назначение у каждой строки кода. Следующим этапом следует выбор метода решения задачи (если в задании метод решения не указан явно) и построение алгоритма. При необходимости алгоритм в целом или его часть изображается в виде блок-схемы или на

	<p>псевдокоде на требуемом уровне детализации.</p> <p>Затем осуществляется кодирование алгоритма на языке C/C++ в виде консольной программы. В ходе написания исходного кода необходимо в текст программы включать только те конструкции языка, которые полностью понятны студенту. Заимствование готовых элементов программы, отдельных инструкций из различных примеров допускается при том же условии.</p> <p>После того, как исходный текст программы готов, выполняется поиск и устранение синтаксических ошибок, после чего выполняется тестирование программы на различных, специально подобранных примерах. Количество и состав тестовых примеров должно быть достаточным для того, чтобы проверить корректность работы программы во всех возможных ситуациях. При обнаружении некорректной работы программы осуществляется ее отладка средствами отладчика; при необходимости выполняется пошаговая трассировка программы; обнаруженные ошибки анализируются и исправляются, после чего проводится повторное тестирование программы.</p> <p>Контроль за выполнением лабораторных работ проходит в виде защиты соответствующей работы преподавателю на компьютере индивидуально каждым студентом. Прием выполненного задания сопровождается устным опросом по затрагиваемым тематическим разделам. Для успешной защиты студент должен:</p> <ul style="list-style-type: none"> - предоставить отлаженную, корректно работающую программу, результаты выполнения которой соответствуют заданию; - быть готовым ответить на вопросы преподавателя, касающиеся: 1) непосредственно исходного кода программы, 2) реализованного в программе алгоритма, 3) теоретического материала, необходимого для выполнения задания; - знать и понимать основные термины предметной области, которой принадлежит решаемая задача; - ответить на вопросы, задаваемые в ходе устного опроса. <p>В случае применения в образовательном процессе дистанционных образовательных технологий обучающиеся выполняют задания на следующих платформах и ресурсах: в команде "MicrosoftTeams". Задания к лабораторным работам размещаются на вкладке Задания канала Общий в соответствующей команде "MicrosoftTeams".</p>
самостоятельная работа	<p>Особенностью обучения бакалавров является высокий уровень самостоятельности обучающихся в ходе образовательного процесса. Можно выделить два вида самостоятельной работы - аудиторная, под руководством преподавателя, и внеаудиторная. Аудиторная самостоятельная работа по дисциплине выполняется на учебных занятиях под непосредственным руководством преподавателя и по его заданию. Внеаудиторная самостоятельная работа выполняется студентом по заданию преподавателя, но без его непосредственного участия.</p> <p>На аудиторных занятиях по дисциплине 'Программирование и алгоритмизация' с участием преподавателя применяются следующие формы СРС:</p> <ul style="list-style-type: none"> - текущие консультации; - разбор и проработка основных приемов работы и способов решения задач по дисциплине. <p>Внеаудиторная СРС по дисциплине:</p> <ul style="list-style-type: none"> - проработка и усвоение теоретического материала на базе рекомендованной преподавателем учебной литературы (включая электронные библиотеки и др.); - подготовка к лабораторным работам (изучение образцов выполнения заданий,

	<p>разобранных примеров решения некоторых задач и др.);</p> <ul style="list-style-type: none"> - оформление отчетов по лабораторным работам; - подготовка к устному опросу; - подготовка к контрольной работе; - подготовка к экзамену. <p>В случае применения в образовательном процессе дистанционных образовательных технологий обучающиеся работают на следующих платформах и ресурсах: в команде "MicrosoftTeams". Все необходимые для самостоятельной работы учебно-методические материалы, учебники, учебные пособия, обучающие видеоролики размещаются на вкладке Файлы канала Общий в соответствующей команде "MicrosoftTeams".</p>
устный опрос	<p>Устный опрос проводится по всем темам (разделам) дисциплины 'Программирование и алгоритмизация'. Опрос проводится на лабораторных занятиях, обучающиеся отвечают на вопросы преподавателя согласно перечню вопросов к устному опросу. Вопросы носят в большинстве случаев теоретический характер и задаются для актуализации лекционного материала, а также для проверки освоения студентом основных терминов, понятий и теоретических положений дисциплины.</p> <p>В случае применения в образовательном процессе дистанционных образовательных технологий обучающиеся работают на следующих платформах и ресурсах: в команде "MicrosoftTeams". Устный опрос проводится в режиме видеособрания в соответствии с расписанием.</p>
контрольная работа	<p>Контрольные работы проводятся в компьютерном классе. Выполнение контрольных работ по дисциплине 'Программирование и алгоритмизация' направлено на проверку овладения навыками разработки и отладки простых консольных алгоритмов в рамках концепций структурного, процедурного программирования. Контрольные работы выполняются студентом в аудиторное время на компьютере в течение 60-80 минут согласно определенному варианту. Текст контрольной работы содержит задачу (задачи), в формулировке которой студенту предлагается написать консольную программу для обработки данных на языке C/C++. Контрольные работы выполняются по следующим темам: 'Массивы', 'Функции'. Выполненные студентом контрольные задания сдаются на проверку в электронном виде.</p> <p>Для успешной защиты контрольной работы студент должен:</p> <ul style="list-style-type: none"> - предоставить отлаженную, корректно работающую программу, результаты выполнения которой соответствуют заданию; - быть готовым ответить на вопросы преподавателя, касающиеся реализованного в программе алгоритма, а также непосредственно исходного кода программы. <p>В случае применения в образовательном процессе дистанционных образовательных технологий обучающиеся выполняют задания на следующих платформах и ресурсах: в команде "MicrosoftTeams". Задания к контрольным работам размещаются на вкладке Задания канала Общий в соответствующей команде "MicrosoftTeams".</p>
экзамен	<p>Экзамен проводится по окончании первого и второго семестров изучения дисциплины. Экзамен является формой промежуточного контроля знаний и умений, полученных на аудиторных занятиях (лекциях, лабораторных работах) и в процессе самостоятельной работы.</p> <p>В ходе подготовки к экзамену студенту рекомендуется проработать теоретический материал лекций, также рекомендуется просмотреть и повторить</p>

	<p>практический материал всех лабораторных работ. В качестве источников получения теоретических и справочных сведений лекции можно рассматривать как первичный, однако не единственный источник. Помимо лекций студент должен активно и самостоятельно работать с литературными источниками, источниками в сети Интернет, в том числе справочными системами и источниками по языку C/C++.</p> <p>Задания, выдаваемые студенту на экзамене, состоят из теоретической и практической частей. Теоретическая часть содержит два вопроса из перечня вопросов к экзамену и требует устного ответа, практическая часть включает задание, выполняемое на компьютере (написать консольную программу для обработки данных на языке C/C++). На подготовку устного ответа и выполнения практического задания студенту дается 1-1,5 часа.</p> <p>Для успешного ответа на экзамене студент должен:</p> <ul style="list-style-type: none"> - корректно и в достаточном объеме осветить данные теоретические вопросы - продемонстрировать знание как лекционного материала, так и материала из литературных источников; - корректно ответить на вопросы, задаваемые в ходе устного опроса по тематике полученных вопросов; - предоставить отлаженную, корректно работающую программу, результаты выполнения которой соответствуют практическому заданию; - ответить на вопросы преподавателя, касающиеся непосредственно технологии выполнения практического задания; - свободно ориентироваться в терминологии тех тем (разделов) дисциплины, к которым принадлежат полученные теоретические вопросы и практическое задание. <p>В случае применения в образовательном процессе дистанционных образовательных технологий обучающиеся работают на следующих платформах и ресурсах: в команде "MicrosoftTeams. Экзамен проводится в режиме видеособрания в соответствии с расписанием.</p> <p>Возможна сдача экзамена по тестам с применением дистанционных технологий в обучении в команде "Microsoft Teams".</p>
--	---

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)

Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем, представлен в Приложении 3 к рабочей программе дисциплины (модуля).

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Материально-техническое обеспечение образовательного процесса по дисциплине (модулю) включает в себя следующие компоненты:

Помещения для самостоятельной работы обучающихся, укомплектованные специализированной мебелью и оснащенные компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду КФУ.

Учебные аудитории – помещения для проведения занятий лекционного типа, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и

промежуточной аттестации, укомплектованные специальной мебелью и оборудованием:

- Компьютеры
- Доска маркерная и меловая
- Интерактивная трибуна с компьютером
- Проектор, экран настенный

Рабочий кабинет – помещение для хранения и профилактического обслуживания учебного оборудования.

12. Средства адаптации преподавания дисциплины к потребностям обучающихся инвалидов и лиц с ограниченными возможностями здоровья

При необходимости в образовательном процессе применяются следующие методы и технологии, облегчающие восприятие информации обучающимися инвалидами и лицами с ограниченными возможностями здоровья:

- создание текстовой версии любого нетекстового контента для его возможного преобразования в альтернативные формы, удобные для различных пользователей;
- создание контента, который можно представить в различных видах без потери данных или структуры, предусмотреть возможность масштабирования текста и изображений без потери качества, предусмотреть доступность управления контентом с клавиатуры;
- создание возможностей для обучающихся воспринимать одну и ту же информацию из разных источников - например, так, чтобы лица с нарушениями слуха получали информацию визуально, с нарушениями зрения - аудиально;
- применение программных средств, обеспечивающих возможность освоения навыков и умений, формируемых дисциплиной, за счёт альтернативных способов, в том числе виртуальных лабораторий и симуляционных технологий;
- применение дистанционных образовательных технологий для передачи информации, организации различных форм интерактивной контактной работы обучающегося с преподавателем, в том числе вебинаров, которые могут быть использованы для проведения виртуальных лекций с возможностью взаимодействия всех участников дистанционного обучения, проведения семинаров, выступления с докладами и защиты выполненных работ, проведения тренингов, организации коллективной работы;
- применение дистанционных образовательных технологий для организации форм текущего и промежуточного контроля;
- увеличение продолжительности сдачи обучающимся инвалидом или лицом с ограниченными возможностями здоровья форм промежуточной аттестации по отношению к установленной продолжительности их сдачи:
 - продолжительности сдачи зачёта или экзамена, проводимого в письменной форме, - не более чем на 90 минут;
 - продолжительности подготовки обучающегося к ответу на зачёте или экзамене, проводимом в устной форме, - не более чем на 20 минут;
 - продолжительности выступления обучающегося при защите курсовой работы - не более чем на 15 минут.

Программа составлена в соответствии с требованиями ФГОС ВО и учебным планом по направлению 01.03.02 "Прикладная математика и информатика"

*Приложение 1
к рабочей программе дисциплины (модуля)
Программирование и алгоритмизация*

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Казанский (Приволжский) федеральный университет»

Набережночелнинский институт (филиал)
Отделение информационных технологий и энергетических систем

Фонд оценочных средств по дисциплине (модулю)
Программирование и алгоритмизация

Направление подготовки: 01.03.02 - Прикладная математика и информатика

Профиль подготовки: отсутствует

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2021

СОДЕРЖАНИЕ

1. СООТВЕТСТВИЕ КОМПЕТЕНЦИЙ ПЛАНИРУЕМЫМ РЕЗУЛЬТАТАМ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)
2. ИНДИКАТОРЫ ОЦЕНИВАНИЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ
3. РАСПРЕДЕЛЕНИЕ ОЦЕНОК ЗА ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНУЮ АТТЕСТАЦИЮ
4. ОЦЕНОЧНЫЕ СРЕДСТВА, ПОРЯДОК ИХ ПРИМЕНЕНИЯ И КРИТЕРИИ ОЦЕНИВАНИЯ
 - 4.1. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ
 - 4.1.1. *Устный опрос*
 - 4.1.1.1. Порядок проведения и процедура оценивания
 - 4.1.1.2. Критерии оценивания
 - 4.1.1.3. Содержание оценочного средства
 - 4.1.2. *Лабораторные работы*
 - 4.1.2.1. Порядок проведения и процедура оценивания
 - 4.1.2.2. Критерии оценивания
 - 4.1.2.3. Содержание оценочного средства
 - 4.1.3. *Контрольная работа*
 - 4.1.3.1. Порядок проведения и процедура оценивания
 - 4.1.3.2. Критерии оценивания
 - 4.1.3.3. Содержание оценочного средства
 - 4.2. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ
 - 4.2.1. *Экзамен (устный/письменный ответ на контрольные вопросы)*
 - 4.2.1.1. Порядок проведения и процедура оценивания
 - 4.2.1.2. Критерии оценивания
 - 4.2.1.3. Оценочные средства
 - 4.2.2. *Экзамен (тестовые задания)*
 - 4.2.2.1. Порядок проведения и процедура оценивания
 - 4.2.2.2. Критерии оценивания
 - 4.2.2.3. Оценочные средства

1. СООТВЕТСТВИЕ КОМПЕТЕНЦИЙ ПЛАНИРУЕМЫМ РЕЗУЛЬТАТАМ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

Код и наименование компетенции	Индикаторы достижения компетенций для данной дисциплины	Оценочные средства текущего контроля и промежуточной аттестации
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	<p>Знать: основные термины и понятия дисциплины программирование и алгоритмизация;</p> <p>Уметь: применять на практике базовые принципы построения алгоритмов, включая принцип последовательной детализации для решения задач профессиональной деятельности;</p> <p>Владеть: навыками разработки и грамотного представления алгоритмов на основе всех основных видов базовых управляющих структур, с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;</p>	<p>Текущий контроль:</p> <p>Семестр 1</p> <p><i>Устный опрос</i></p> <p>Темы:</p> <p>1. Информационно-логические основы работы ЭВМ</p> <p>2. Алгоритмы и алгоритмизация</p> <p>3. Проектирование алгоритмов и программ</p> <p>4. Лексика языка C++. Структура программы</p> <p>5. Массивы, символьные строки и структуры</p> <p>Семестр 2</p> <p><i>Устный опрос</i></p> <p>Темы:</p> <p>6. Функции</p> <p>7. Файловый ввод-вывод</p> <p>8. Указатели и динамическое распределение памяти. Ссылки</p> <p>Промежуточная аттестация:</p> <p>Семестр 1</p> <p><i>Экзамен (вопросы к экзамену)</i></p> <p>Семестр 2</p> <p><i>Экзамен (вопросы к экзамену)</i></p>
ПК-7 Способен к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения	<p>Знать: существующие подходы к проектированию и методы проектирования алгоритмов и программ в области системного и прикладного программного обеспечения;</p> <p>Уметь: создавать простые программы на языке программирования C/C++ в среде Visual Studio.NET или аналогичной в рамках идеологии структурного и процедурного программирования;</p> <p>Владеть: навыками разработки простых консольных алгоритмов в рамках концепций структурного, процедурного</p>	<p>Текущий контроль:</p> <p>Семестр 1</p> <p><i>Лабораторные работы</i></p> <p>Темы:</p> <p>4. Лексика языка C++. Структура программы</p> <p>5. Массивы, символьные строки и структуры</p> <p><i>Контрольная работа</i></p> <p>Темы:</p> <p>5. Массивы, символьные строки и структуры</p> <p>Семестр 2</p> <p><i>Лабораторные работы</i></p> <p>Темы:</p> <p>6. Функции</p> <p>7. Файловый ввод-вывод</p> <p>8. Указатели и динамическое распределение памяти. Ссылки</p>

	программирования;	<i>Контрольная работа</i> Темы: 5. Функции Промежуточная аттестация: Семестр 1 <i>Экзамен (вопросы к экзамену)</i> Семестр 2 <i>Экзамен (вопросы к экзамену)</i>
ПК-8 Способен к разработке стратегии тестирования и управлению процессом тестирования	Знать: понятие интегрированной среды программирования, назначение ее компонентов, последовательность этапов создания приложения, включая процесс тестирования, с использованием языков программирования; Уметь: организовывать консольный ввод-вывод в программах на языке C/C++, разрабатывать стратегии тестирования программы, управлять процессом тестирования; Владеть: навыками самостоятельной работы с литературой, иными источниками информации по дисциплине.	Текущий контроль: Семестр 1 <i>Лабораторные работы</i> Темы: 4. Лексика языка C++. Структура программы 5. Массивы, символьные строки и структуры <i>Контрольная работа</i> Темы: 5. Массивы, символьные строки и структуры Семестр 2 <i>Лабораторные работы</i> Темы: 6. Функции 7. Файловый ввод-вывод 8. Указатели и динамическое распределение памяти. Ссылки <i>Контрольная работа</i> Темы: 5. Функции Промежуточная аттестация: Семестр 1 <i>Экзамен (вопросы к экзамену)</i> Семестр 2 <i>Экзамен (вопросы к экзамену)</i>

2. ИНДИКАТОРЫ ОЦЕНИВАНИЯ СФОРМИРОВАННОСТИ КОМПЕТЕНЦИЙ

Компетенция	Зачтено			Не зачтено
	Высокий уровень (отлично) (86-100 баллов)	Средний уровень (хорошо) (71-85 баллов)	Низкий уровень (удовлетворительно) (56-70 баллов)	Ниже порогового уровня (неудовлетворительно) (0-55 баллов)
ОПК-5	Знает все основные термины и понятия дисциплины, их содержание и взаимосвязь	Знает большинство терминов и понятий дисциплины, их содержание	Знает некоторые термины и понятия дисциплины, их содержание	Не знает основных терминов и понятий дисциплины

	Умеет разрабатывать и грамотно представлять алгоритмы на основе всех основных видов базовых управляющих структур с применением принципа последовательной детализации	Умеет разрабатывать и представлять алгоритмы на основе основных видов базовых управляющих структур с применением принципа последовательной детализации	Умеет разрабатывать и представлять алгоритмы на основе некоторых отдельных видов базовых управляющих структур	Не умеет применять на практике базовые принципы построения алгоритмов
	Владеет в полной мере навыками самостоятельной работы с литературой, иными источниками информации по дисциплине	Владеет навыками самостоятельной работы с литературой, иными источниками информации по дисциплине	Частично владеет навыками самостоятельной работы с литературой, иными источниками информации по дисциплине	Не владеет навыками самостоятельной работы с литературой, иными источниками информации по дисциплине
ПК-7	Знает все существующие подходы к проектированию и методы проектирования алгоритмов и программ в рамках объема дисциплины	Знает большинство подходов к проектированию и методов проектирования алгоритмов и программ в рамках объема дисциплины	Знает некоторые из подходов и методов проектирования алгоритмов и программ	Не знает большинства подходов и методов проектирования алгоритмов и программ в рамках объема дисциплины
	Умеет создавать программы на языке программирования C/C++ в среде VisualStudio.NET или аналогичной в рамках идеологии структурного и процедурного программирования	Умеет с небольшими недостатками создавать программы на языке программирования C/C++ в среде VisualStudio.NET или аналогичной в рамках идеологии структурного и процедурного программирования	Частично умеет создавать программы на языке программирования C/C++ в среде VisualStudio.NET или аналогичной в рамках идеологии структурного программирования	Не умеет создавать программы на языке программирования C/C++ в среде VisualStudio.NET или аналогичной
	Владеет навыками разработки консольных алгоритмов в рамках концепций структурного, процедурного программирования	Владеет с небольшими недостатками навыками разработки консольных алгоритмов в рамках концепций структурного,	Частично владеет навыками разработки консольных алгоритмов в рамках концепций структурного программирования	Не владеет навыками разработки консольных алгоритмов в рамках концепций структурного, процедурного программирования

		процедурного программирования		
ПК-8	Знает понятие интегрированной среды программирования, все ее компоненты, их назначение, представляет последовательность всех этапов создания приложения	Знает понятие интегрированной среды программирования, большинство ее компоненты, их назначение, представляет последовательность этапов создания приложения	Знает понятие интегрированной среды программирования, частично знает ее компоненты, их назначение, частично представляет последовательность этапов создания приложения	Практически не знает понятие интегрированной среды программирования, ее компоненты, плохо представляет последовательность этапов создания приложения
	Умеет организовывать консольный ввод-вывод в программах на языке C/C++, правильно тестировать разработанную программу, адекватно оценивать результаты ее работы	Умеет организовывать консольный ввод-вывод в программах на языке C/C++, имеет представление о подборке тестовых примеров и оценке результатов работы программы	Частично умеет организовывать консольный ввод-вывод в программах на языке C/C++, имеет затруднения в оценке результатов работы программы	Не умеет организовывать консольный ввод-вывод в программах на языке C/C++, имеет затруднения в оценке результатов работы программы
	Владеет навыками работы в программной среде, грамотного представления алгоритмов на основе основных видов базовых управляющих структур	С небольшими недочетами владеет навыками работы в программной среде, грамотного представления алгоритмов на основе основных видов базовых управляющих структур	Частично владеет навыками работы в программной среде, удовлетворительно разрабатывает алгоритмы на основе основных видов базовых управляющих структур	Не владеет навыками работы в программной среде, грамотного представления алгоритмов на основе основных видов базовых управляющих структур

3. РАСПРЕДЕЛЕНИЕ ОЦЕНОК ЗА ФОРМЫ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНУЮ АТТЕСТАЦИЮ

Семестр 1:

Текущий контроль:

Устный опрос – 10 баллов

Лабораторные работы – 20 баллов

Контрольная работа – 20 баллов

Итого 10+20+20 = 50 баллов

Промежуточная аттестация – экзамен.

Экзамен является формой промежуточного контроля знаний и умений, полученных на аудиторных занятиях (лекциях, лабораторных работах) и в процессе самостоятельной работы. Экзамен проводится в устной или письменной форме по билетам. Билет содержит два вопроса из перечня вопросов к экзамену. На подготовку ответа студенту

дается 1-1,5 часа. Всего вопросов – 34.

Экзамен – 50 баллов

Общее количество баллов по дисциплине за текущий контроль и промежуточную аттестацию: 50+50=100 баллов.

Семестр 2:

Текущий контроль:

Устный опрос – 10 баллов

Лабораторные работы – 30 баллов

Контрольная работа – 10 баллов

Итого 10+30+10 = 50 баллов

Промежуточная аттестация – экзамен.

Экзамен является формой промежуточного контроля знаний и умений, полученных на аудиторных занятиях (лекциях, лабораторных работах) и в процессе самостоятельной работы. Экзамен проводится в устной или письменной форме по билетам. Билет содержит два вопроса из перечня вопросов к экзамену. На подготовку ответа студенту дается 1-1,5 часа. Всего вопросов – 30.

Экзамен – 50 баллов

При проведении экзамена в форме тестирования: в тесте 30 вопросов, баллы за тест - 50 баллов. Тестовые задания загружаются в Microsoft Teams заранее. При проведении экзамена в форме тестирования студентам дается 60 минут.

Общее количество баллов по дисциплине за текущий контроль и промежуточную аттестацию: 50+50=100 баллов.

Соответствие баллов и оценок:

Для экзамена:

86-100 – отлично

71-85 – хорошо

56-70 – удовлетворительно

0-55 – неудовлетворительно

4.1. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ

4.1.1. Устный опрос

4.1.1.1. Порядок проведения и процедура оценивания

Устный опрос проводится на практических занятиях. Обучающиеся выступают с докладами, сообщениями, дополнениями, участвуют в дискуссии, отвечают на вопросы преподавателя. Оценивается уровень домашней подготовки по теме, способность системно и логично излагать материал, анализировать, формулировать собственную позицию, отвечать на дополнительные вопросы.

4.1.1.2. Критерии оценивания

1) 86-100% от максимального числа баллов

- обучающийся знает весь теоретический материал по рассматриваемому вопросу, предусмотренный учебной программой;

- корректно использует понятийный аппарат;

- высказывает свою точку зрения по рассматриваемому вопросу и может ее аргументированно обосновать.

2) 71-85% от максимального числа баллов

- обучающийся знает основные теоретические положения по рассматриваемому

вопросу;

- корректно использует понятийный аппарат;
 - высказывает свою точку зрения.
- 3) 56-70% от максимального числа баллов
- обучающийся имеет общее представление о предмете обсуждения;
 - допускает ошибки при использовании понятийного аппарата;
 - высказывает свои мысли сумбурно, ответ слабо структурирован.
- 4) 0-55% от максимального числа баллов
- обучающийся не владеет теоретическим материалом;
 - не владеет понятийным аппаратом;
 - не способен внятно сформулировать свои мысли.

4.1.1.3. Содержание оценочного средства

Семестр 1

Тема 1. Информационно-логические основы работы ЭВМ

Устройство персонального компьютера, назначение микропроцессора, оперативной памяти; устройства ввода-вывода; логическая структура компьютера; представление числовой и символьной информации в ЭВМ; виды и назначение систем счисления.

Тема 2. Алгоритмы и алгоритмизация

Понятия алгоритма, исполнителя алгоритма, системы команд исполнителя; свойства алгоритма с иллюстрацией их на примере алгоритма Евклида (или иного); этапы разработки программного обеспечения для ЭВМ; понятия ввода и вывода данных.

Тема 3. Проектирование алгоритмов и программ

Три основных подхода к проектированию алгоритмов и программ; структурное проектирование алгоритмов, базовые управляющие структуры; средства для создания приложений; понятие языка программирования, их классификация.

Тема 4. Лексика языка C++. Структура программы

Процесс создания программы; идентификаторы и типы данных; функции и данные; главная функция программы; выражение; инструкция присваивания; виды и синтаксис базовых управляющих структур в языке C/C++.

Тема 5. Массивы, символьные строки и структуры

Понятие массива, виды массивов и их объявление; понятие символьных и строковых данных; C-строки и действия над ними; понятие структуры и их назначение.

Семестр 2

Тема 6. Функции

Подпрограмма как основной элемент программы в процедурном программировании; назначение подпрограмм, преимущества их использования в программировании; определение и вызов функций в C/C++; формальные и фактические аргументы функций; способы возврата результата из функций; передача массивов в функции через формальные аргументы.

Тема 7. Файловый ввод-вывод

Понятия файла, файлового ввода-вывода; бинарные и текстовые файлы; файловая переменная; основные этапы реализации файлового ввода-вывода; основные функции для файлового ввода вывода в стиле языка C.

Тема 8. Указатели и динамическое распределение памяти. Ссылки

Понятие указателя; логическая структура оперативной памяти, адресное пространство; принцип динамического выделения памяти; тип ссылки, его отличие от типа указателя. Применение указателей и ссылок. Проблемы, возникающие при использовании указателей: утечка памяти, неинициализированные указатели.

4.1.2. Лабораторные работы

4.1.2.1. Порядок проведения и процедура оценивания

Лабораторные занятия проводятся в компьютерном классе. Задания по лабораторным работам, рекомендации по их выполнению и примеры выполнения находятся в соответствующих методических материалах. В качестве общих рекомендаций по выполнению лабораторных заданий следует отметить следующее.

Перед началом выполнения задания желательно проработать соответствующий теоретический материал, внимательно изучить иллюстрирующие методические примеры. Далее студент должен детально разобрать пример решения схожего выполняемого задания (если есть), внимательно проанализировать программный код. В результате студент должен сформировать максимально четкое и конкретное представление о том, как выполняется программа в примере, какой смысл и назначение у каждой строки кода.

Следующим этапом следует выбор метода решения задачи (если в задании метод решения не указан явно) и построение алгоритма. При необходимости алгоритм в целом или его часть изображается в виде блок-схемы или на псевдокоде на требуемом уровне детализации.

Затем осуществляется кодирование алгоритма на языке C/C++ в виде консольной программы. В ходе написания исходного кода необходимо в текст программы включать только те конструкции языка, которые полностью понятны студенту. Заимствование готовых элементов программы, отдельных инструкций из различных примеров допускается при том же условии.

После того, как исходный текст программы готов, выполняется поиск и устранение синтаксических ошибок, после чего выполняется тестирование программы на различных, специально подобранных примерах. Количество и состав тестовых примеров должно быть достаточным для того, чтобы проверить корректность работы программы во всех возможных ситуациях. При обнаружении некорректной работы программы осуществляется ее отладка средствами отладчика; при необходимости выполняется пошаговая трассировка программы; обнаруженные ошибки анализируются и исправляются, после чего проводится повторное тестирование программы.

Контроль за выполнением лабораторных работ проходит в виде защиты соответствующей работы преподавателю на компьютере индивидуально каждым студентом. Прием выполненного задания сопровождается устным опросом по затрагиваемым тематическим разделам. Для успешной защиты студент должен:

- предоставить отлаженную, корректно работающую программу, результаты выполнения которой соответствуют заданию;
- быть готовым ответить на вопросы преподавателя, касающиеся: 1) непосредственно исходного кода программы, 2) реализованного в программе алгоритма, 3) теоретического материала, необходимого для выполнения задания;
- знать и понимать основные термины предметной области, которой принадлежит решаемая задача;
- ответить на вопросы, задаваемые в ходе устного опроса.

В случае применения в образовательном процессе дистанционных образовательных технологий обучающиеся выполняют задания на следующих платформах и ресурсах: в команде "MicrosoftTeams". Задания к лабораторным работам размещаются на вкладке Задания канала Общий в соответствующей команде "MicrosoftTeams".

4.1.2.2. Критерии оценивания

Баллы в интервале 86-100% от максимальных ставятся, если обучающийся:

Оборудование и методы использованы правильно. Владеет теоретическим материалом в полном объеме, использует оптимальные методы. Необходимые навыки и

умения полностью освоены. Результат лабораторной работы полностью соответствует её целям.

Баллы в интервале 71-85% от максимальных ставятся, если обучающийся:

Оборудование и методы использованы в основном правильно. Владеет теоретическим материалом, использует подходящие методы. Необходимые навыки и умения в основном освоены. Результат лабораторной работы в основном соответствует её целям.

Баллы в интервале 56-70% от максимальных ставятся, если обучающийся:

Оборудование и методы частично использованы правильно. Владеет только базовыми понятиями, не всегда использует подходящие методы. Необходимые навыки и умения частично освоены. Результат лабораторной работы частично соответствует её целям, задание выполнено с ошибками.

Баллы в интервале 0-55% от максимальных ставятся, если обучающийся:

Оборудование и методы использованы неправильно. Не владеет даже базовыми понятиями. Необходимые навыки и умения не освоены. Результат лабораторной работы не соответствует её целям, допущены грубые ошибки.

4.1.2.3. Содержание оценочного средства

Семестр 1

Лабораторная работа № 1. Простые вычисления по формулам

Составление алгоритма и программы вычисления и вывода на экран заданной формулы в виде функции $f(x)$, где x задан на некотором отрезке области определения функции.

Лабораторная работа № 2. Накопление сумм и произведений в цикле

Разработка циклического алгоритма и программы для вычисления результата по формуле с накоплением суммы (или произведения) и вывода на экран.

Лабораторная работа № 3. Вычисление корня функционального уравнения

Разработка алгоритма и программы для решения заданного нелинейного уравнения общего вида $f(x)=0$ методом деления отрезка пополам и вывода полученного результата на экран.

Лабораторная работа № 4. Табулирование функции с выводом результата по логическому условию

Разработка алгоритма и программы для вычисления значений функции $f(x)$ на заданном интервале для тех значений аргумента, которые удовлетворяют заданному сложному условию и подсчета количества остальных ее значений.

Лабораторная работа № 5. Логическая проверка попадания точки в заданную область

Разработка алгоритма и программы для проверки попадания каждой из набора точек на плоскости в некоторую область, подсчета количества попавших в область точек и вывода их координат на экран.

Лабораторная работа № 6. Сложное условие завершения цикла

Разработка алгоритма и программы для вычисления значений функции $f(x)$ и вывода на экран результатов до тех пор, пока не будет найдена некоторая характерная точка графика.

Лабораторная работа № 7. Обработка одномерных массивов с единственным циклом

Разработка алгоритма и программы с единственным циклом для обработки одномерных массивов и вывода результатов обработки на экран.

Лабораторная работа № 8. Сложное условие завершения цикла при работе с матрицами

Разработка алгоритма и программы для обработки числовых данных, представленных в виде двумерного массива, в которой предусмотреть возможность

остановки процесса обработки данных при выполнении некоторого заданного условия.

Лабораторная работа № 9. Обработка матриц произвольного размера

Разработка алгоритма и программы для выполнения некоторой обработки числовых данных, представленных в виде двумерного массива, и вывода результата обработки на экран.

Лабораторная работа № 10 (дополнительная). Обработка символьных строк.

Семестр 2

Лабораторная работа №1. Вычисление выражения с одной функцией

Составить программу, использующую функцию для организации вычислений по заданной формуле. Для этого следует выявить в формуле выражения, подобные по своей схеме, т.е. различающиеся только объектами операций (операндами). Разработанная для вычислений функция должна использовать только свои локальные переменные и формальные аргументы.

Лабораторная работа №2. Использование массивов в качестве аргументов функции

Написать программу, использующую подпрограмму (функцию), аргументами которой являются либо два одномерных массива (или один из них), либо матрица. Результат, возвращаемый подпрограммой, указан в вариантах индивидуальных заданий. Для работы подпрограмма должна использовать только свои локальные переменные и формальные аргументы. В главной функции программы должен осуществляться ввод исходных данных, однократный вызов подпрограммы обработки данных и вывод на экран.

Лабораторная работа №3. Разработка программы с использованием созданной по условию задачи функции

Составить программу, в которой используется подпрограмма (функция), разработанная по условию индивидуального задания. Для работы подпрограмма должна использовать только свои локальные переменные и формальные аргументы.

Лабораторная работа №4. Вычисление заданной функции путем разложения её в ряд с использованием рекурсии

Составить программу вычисления заданной математической функции путем разложения её в ряд. Программа должна применять рекурсивную функцию для вычисления суммы ряда с заданной точностью. Для оценки правильности вычисления суммы ряда в программе следует также предусмотреть вычисление математической функции по контрольной формуле (полученное приближенное значение функции (сумма ряда) и значение, вычисленное по контрольной формуле, не должны различаться более, чем на величину заданной точности).

Лабораторная работа №5. Разработка многомодульной программы с выделением подпрограмм

Разработать программу, выполняющую обработку данных согласно варианту индивидуального задания. В программе для решения задачи разработать, исходя из целесообразности, подпрограммы (одну или несколько). Все созданные подпрограммы должны во время выполнения пользоваться только собственными локальными переменными и формальными аргументами.

Лабораторная работа №6 (дополнительная). Вычисление корня функционального уравнения при разных значениях точности

Лабораторная работа №7. Обработка символьных строк

Разработать программу, выполняющую чтение из произвольного текстового файла символьных строк и их обработку согласно варианту индивидуального задания. Необходимо вывести на экран исходный текст и результат его обработки.

Предполагается, что строки текста состоят из слов. Словом считается набор символов (как правило, только букв и цифр), не содержащий разделителей. В качестве разделителей следует рассматривать символы "пробел", "запятая", "точка". Также дополнительно можно рассматривать в качестве разделителей другие символы,

например, символ "точка с запятой", символ табуляции, символ перехода на следующую строку, символ возврата каретки и пр.

Лабораторная работа №8. Файловый ввод-вывод, ввод с клавиатуры и обработка массива записей

Разработать программу, выполняющую создание в памяти динамического массива структур, вывод текущих данных в виде таблицы на экран, обработку данных согласно индивидуальному заданию и уничтожение массива данных из памяти перед завершением программы. Исходные данные о структуре элементов массива приведены в вариантах задания.

В программе реализовать (в виде отдельных функций) следующие операции над данными: чтение данных из файла, сохранение данных из файла, добавление новой записи к списку данных, изменение записи с заданным номером, поиск записей, удовлетворяющих заданному условию.

Программа должна быть составлена таким образом, чтобы все операции над данными могли выбираться пользователем в произвольном порядке по номеру операции, вводимому им с клавиатуры. Каждая операция должна быть реализована в программе в виде отдельной функции. Операции файлового ввода-вывода должны выполняться только при выборе пользователем соответствующих команд (Загрузить из файла, Сохранить в файл).

Лабораторная работа №9. Обработка линейного связного списка структур с применением стандартных средств консольного ввода-вывода

Разработать программу, выполняющую создание в памяти связного списка структур, вывод созданного списка на экран, обработку списка согласно индивидуальному заданию и уничтожение его из памяти перед завершением программы. При этом следует использовать указатели и динамическое распределение памяти. Исходные данные о структуре узлов списка приведены в вариантах задания.

Во время своей работы программа должна выполнять над связным списком следующие операции: ввод списка с клавиатуры; вывод списка на экран; добавление новых узлов в начало списка, в его конец или в заданную позицию списка; удаление из списка узла по его номеру; сохранение в файле и чтение из файла всего списка; определение длины списка; обработку списка согласно индивидуальному заданию.

Программа должна быть составлена таким образом, чтобы все операции над связным списком могли выбираться пользователем в произвольном порядке по номеру операции, вводимому им с клавиатуры. Каждая операция должна быть реализована в программе в виде отдельной функции. Операции файлового ввода-вывода должны выполняться только при выборе пользователем соответствующих команд (Загрузить из файла, Сохранить в файл).

Лабораторная работа №10 (дополнительная). Обработка линейного связного списка структур с применением нестандартных средств консольного ввода-вывода

4.1.2. Контрольная работа

4.1.3.1. Порядок проведения и процедура оценивания

Контрольные работы проводятся в компьютерном классе. Выполнение контрольных работ по дисциплине 'Программирование и алгоритмизация' направлено на проверку овладения навыками разработки и отладки простых консольных алгоритмов в рамках концепций структурного, процедурного программирования. Контрольные работы выполняются студентом в аудиторное время на компьютере в течение 60-80 минут согласно определенному варианту. Текст контрольной работы содержит задачу (задачи), в формулировке которой студенту предлагается написать консольную программу для обработки данных на языке C/C++. Контрольные работы выполняются по следующим темам: 'Массивы', 'Функции'. Выполненные студентом контрольные задания сдаются на

проверку в электронном виде.

Для успешной защиты контрольной работы студент должен:

- предоставить отлаженную, корректно работающую программу, результаты выполнения которой соответствуют заданию;
- быть готовым ответить на вопросы преподавателя, касающиеся: реализованного в программе алгоритма, а также непосредственно исходного кода программы.

В случае применения в образовательном процессе дистанционных образовательных технологий обучающиеся выполняют задания на следующих платформах и ресурсах: в команде "MicrosoftTeams". Задания к контрольным работам размещаются на вкладке Задания канала Общий в соответствующей команде "MicrosoftTeams".

4.1.3.2. Критерии оценивания

Баллы в интервале 86-100% от максимальных ставятся, если обучающийся:

Правильно выполнены все задания. Продемонстрировано владение всем теоретическим материалом. Проявлена способность применять знания и умения к выполнению конкретных заданий, используются оптимальные методы.

Баллы в интервале 71-85% от максимальных ставятся, если обучающийся:

Правильно выполнена большая часть заданий. Присутствуют незначительные ошибки. Продемонстрирован уровень владения значительной частью материала. Проявлены способности применять знания и умения к выполнению конкретных заданий, используются подходящие методы.

Баллы в интервале 56-70% от максимальных ставятся, если обучающийся:

Задания выполнены более чем наполовину. Присутствуют серьёзные ошибки. Продемонстрировано владение базовыми положениями теоретического материала. Проявлены низкие способности применять знания и умения к выполнению конкретных заданий, не всегда используются подходящие методы.

Баллы в интервале 0-55% от максимальных ставятся, если обучающийся:

Задания выполнены менее чем наполовину. Обучающийся не владеет даже базовыми понятиями, неправильно использует методы, допускает грубые ошибки.

4.1.3.3. Содержание оценочного средства

Семестр 1

Тема 5. Массивы, символьные строки и структуры

Написать на языке C++ и отладить консольную программу решения задачи обработки одномерных массивов, для вычислений используя единственный цикл, согласно индивидуальному заданию.

Указания:

- В программе необходимо предусмотреть возможность ввода пользователем размера используемых массивов.
- Ввод исходных данных в программе следует выполнять до начала вычислений, а вывод результата - после завершения вычислений.

Варианты индивидуальных заданий:

1. Задано n точек на плоскости, абсциссы и ординаты которых заданы в массивах X и Y соответственно. Необходимо определить, у какого числа этих точек положительна как абсцисса, так и ордината, а также найти среднюю ординату всех остальных точек из числа заданных.

2. Вычислить куб суммы и количество тех элементов заданного массива X , значения которых меньше заданного числа g или находятся в заданном интервале $[a, b]$.

3. Для заданной величины g и заданных массивов X и Y определить количество произведений вида $x[i]*y[i]$, удовлетворяющих условию $x[i]*y[i] < g$, и сумму таких произведений.

4. Для заданного массива X найти по отдельности суммы ($S1$ и $S2$) и количества ($N1$ и $N2$) элементов, значения которых соответственно больше W и меньше W , где W - заданное положительное число.

5. Найти среднее арифметическое тех элементов заданного одномерного массива X , значения которых не превышают $x[1]$, включая и сам элемент $x[1]$. Найти также среднее арифметическое всех элементов данного массива.

6. Вычислить выражение $CX * CY$, где CX и CY - средние арифметические положительных элементов заданных массивов $X = \{x_1, x_2, \dots, x_n\}$ и $Y = \{y_1, y_2, \dots, y_n\}$ соответственно.

7. Для заданного массива $X = \{x_1, x_2, \dots, x_n\}$ найти среднее арифметическое CX элементов, имеющих четные номера, и при том положительных, а для заданного массива $Y = \{y_1, y_2, \dots, y_n\}$ найти среднее арифметическое CY элементов, имеющих нечетные номера, и притом отрицательных.

8. Заданы целочисленные массивы $X = \{x_1, x_2, \dots, x_n\}$ и $Y = \{y_1, y_2, \dots, y_n\}$. Заменить значение каждого элемента массива Y новым значением, определяемым по правилу $y_i = x_i - y_i$ ($i = 1 \dots n$), и подсчитать число случаев, когда выполняется равенство $x_i = y_i$.

9. Найти сумму и общее количество тех элементов заданного массива $X = \{x_1, x_2, \dots, x_n\}$, абсолютная величина (модуль) которых отличается от заданного значения p не более, чем на величину t .

10. Для заданных целочисленных массивов $X = \{x_1, x_2, \dots, x_n\}$ и $Y = \{y_1, y_2, \dots, y_n\}$, проверяя на равенство элементы пар (x_1, y_1) , (x_2, y_2) , ..., (x_n, y_n) подсчитать количество случаев равенства элементов пары; одновременно найти среднее арифметическое элементов массива X .

Семестр 2

Тема 6. Функции

Написать на языке C++ и отладить консольную программу решения задачи. Разработать программу, выполняющую обработку данных согласно варианту индивидуального задания. В программе для решения задачи разработать, исходя из целесообразности, функции (одну или несколько). Все созданные подпрограммы должны при выполнении пользоваться только собственными локальными переменными и формальными аргументами.

Варианты задания:

1. В прямоугольной матрице A в каждом столбце поставить на первое место максимальный элемент столбца. И, если среди полученных элементов первой строки не окажется элементов, по модулю меньших заданной величины S , разделить элементы последней строки на соответствующие элементы первой строки.

2. Если первая строка прямоугольной матрицы A имеет максимальное количество отрицательных элементов, проверить, как изменится среднее арифметическое всей матрицы, если заменить все отрицательные элементы матрицы их модулями.

3. Если в прямоугольной матрице A все суммы элементов строк попадают на заданный отрезок $[b, c]$, определить номер строки с максимальной суммой элементов, иначе определить номера строк, сумма элементов которых не попала на отрезок $[b, c]$.

4. Определить столбец прямоугольной матрицы A с максимальной суммой элементов b , если его номер больше заданного, сформировать новую матрицу B из столбцов исходной до найденного столбца, иначе скопировать в вектор Y элементы заданного столбца.

5. Если заданная квадратная целочисленная матрица A является треугольной (элементы выше главной диагонали равны нулю), вычислить ее среднее арифметическое, иначе определить, сколько элементов, лежащих выше главной диагонали, отличны от нуля.

6. Переставить в каждом столбце прямоугольной матрицы A размером n строк и m

столбцов все отрицательные элементы в конец столбца. Вывести на экран часть полученной матрицы, состоящую из s первых строк, не имеющих отрицательных элементов ($1 < s < n$).

7. Если все точки плоскости, заданные своими координатами, попадают в круг с радиусом r и центром в начале координат, определить их среднюю абсциссу и ординату, иначе вывести на экран номера точек, не попавших в заданный круг.

8. Если максимальный элемент квадратной матрицы A находится выше главной диагонали, транспонировать матрицу, сохранив ее в отдельном массиве B , иначе определить сумму элементов строки и столбца с номерами, равными индексам максимального элемента.

9. Дана квадратная матрица A . Увеличить все элементы строки с минимальной суммой элементов на среднее арифметическое элементов матрицы, лежащих выше главной диагонали.

10. Изменить заданную прямоугольную матрицу A так, чтобы на первом месте стояла строка с максимальной, а на последнем месте строка с минимальной суммой элементов, сохранив все элементы исходной матрицы.

4.2. ОЦЕНОЧНЫЕ СРЕДСТВА ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

4.2.1. Экзамен (устный/письменный ответ на контрольные вопросы)

4.2.1.1. Порядок проведения и процедура оценивания

Экзамен проводится по окончании первого и второго семестров изучения дисциплины. Экзамен является формой промежуточного контроля знаний и умений, полученных на аудиторных занятиях (лекциях, лабораторных работах) и в процессе самостоятельной работы.

В ходе подготовки к экзамену студенту рекомендуется проработать теоретический материал лекций, также рекомендуется просмотреть и повторить практический материал всех лабораторных работ. Лекционный материал доступен в составе электронного учебно-методического комплекса по дисциплине, размещенного на сервере локальной сети института. В качестве источников получения теоретических и справочных сведений лекции можно рассматривать как первичный, однако не единственный источник. Помимо лекций студент должен активно и самостоятельно работать с литературными источниками, источниками в сети Интернет, в том числе справочными системами и источниками по языку C/C++. Приветствуется знание технического английского языка, достаточное для чтения и понимания технических справочных текстов.

Для успешного ответа на экзамене студент должен:

- корректно и в достаточном объеме осветить данные теоретические вопросы - продемонстрировать знания как лекционного материала, так и материала из литературных источников;
- корректно ответить на вопросы, задаваемые в ходе устного опроса по тематике полученных вопросов;
- свободно ориентироваться в терминологии тех тем (разделов) дисциплины, к которым принадлежат полученные теоретические вопросы.

В случае применения в образовательном процессе дистанционных образовательных технологий обучающиеся работают на следующих платформах и ресурсах: в команде "MicrosoftTeams". Экзамен проводится в режиме видеособрания в соответствии с расписанием.

4.2.1.2. Критерии оценивания

Баллы в интервале 86-100% от максимальных ставятся, если:

Обучающийся дал полный ответ на все вопросы, при ответе использовал примеры практического применения рассматриваемого теоретического материала, ответил на все

дополнительные вопросы, ответ четкий и хорошо структурированный, освоен понятийный аппарат.

Баллы в интервале 71-85% от максимальных ставятся, если:

Обучающийся дал полный ответ на все вопросы, однако испытывал затруднение с приведением практических примеров применения рассматриваемого теоретического материала, ответил не на все дополнительные вопросы, ответ структурирован, освоен понятийный аппарат.

Баллы в интервале 56-70% от максимальных ставятся, если:

Обучающийся раскрыл вопросы лишь частично, не смог привести практические примеры применения рассматриваемого теоретического материала, частично ответил на некоторые из дополнительных вопросов, допускает несущественные ошибки при использовании понятийного аппарата.

Баллы в интервале 0-55% от максимальных ставятся, если:

Обучающийся не ответил на вопросы или же ответы не соответствовали заданным вопросам, не дал адекватного ответа на дополнительные вопросы, допускает грубые ошибки при использовании понятийного аппарата или не использует понятийный аппарат предметной области вовсе.

4.2.1.3. Оценочные средства

Вопросы к экзамену

Семестр 1

1. Системы счисления. Способы перевода чисел из одной системы счисления в другую.
2. Способы представления чисел в ЭВМ. Формат с фиксированной запятой. Формат с плавающей запятой.
3. Способы представления символьных данных в ЭВМ. Кодировки.
4. Этапы подготовки и решения задач по разработке программ для ЭВМ.
5. Понятие алгоритма. Свойства. Пример.
6. Способы представления алгоритмов. Достоинства и недостатки способов.
7. Восходящее проектирование алгоритмов. Нисходящее проектирование алгоритмов.
8. Идеологии проектирования алгоритмов и программ.
9. Базовые управляющие структуры: следование, развилки. Теорема о структурировании.
10. Базовые управляющие структуры: следование, циклы. Теорема о структурировании.
11. Классификация языков программирования.
12. Область применения языков низкого уровня.
13. Понятие и структура интегрированной среды программирования (IDE).
14. Процесс создания приложения. Трансляторы. Компоновщик. Отладчик.
15. Процесс создания приложения. Виды ошибок в программе. Пошаговая отладка программы.
16. Лексика языка C++. Понятие идентификатора. Правила создания идентификаторов.
17. Лексика языка C++. Литералы. Операторы. Комментарии. Директивы препроцессора.
18. Понятие о данных программы и их типах. Понятие переменной, константы.
19. Обзор фундаментальных типов данных. Встроенные типы.
20. Объявление и определение переменных и констант. Инициализация. Примеры.
21. Понятие ввода и вывода данных. Консольный ввод/вывод в стиле C. Спецификации преобразования. Примеры.

22. Консольный ввод/вывод в стиле C++. Объекты потоков ввода/вывода. Манипуляторы.

23. Инструкция присваивания. Примеры. Понятие выражения.

24. Структура консольной программы на C/C++. Процесс создания готовой программы.

25. Выражения. Структура выражения. Арифметические и логические выражения. Именуемые выражения. Понятие операции. Примеры выражений.

26. Операторы в выражениях. Приоритет операций. Обзор операторов языка C++. Примеры.

27. Базовые управляющие структуры в C++. Понятие блока. Развилки.

28. Развилка с множественным выбором switch-case

29. Базовые управляющие структуры в C++. Понятие блока. Цикл с предусловием.

30. Базовые управляющие структуры в C++. Понятие блока. Цикл с постусловием.

31. Базовые управляющие структуры в C++. Понятие блока. Цикл с параметром.

32. Понятие о массиве. Виды массивов. Структура массива в памяти. Объявление массива. Доступ к элементам массива.

33. Понятие о строковых данных. С-строка: понятие, структура, основные принципы работы с ней. Пример.

34. Обзор основных операций над строками. Стандартная библиотека C для строковых данных.

Семестр 2

1. Понятие о подпрограммах. Назначение подпрограмм.

2. Идеология процедурного программирования.

3. Взаимодействие подпрограмм с программой. Вызов подпрограммы.

4. Локальные и глобальные переменные. Области видимости переменных.

5. Пространства имен.

6. Локальные и глобальные переменные. Области видимости переменных.

7. Понятие и назначение одномерных массивов, принципы работы с ними, особенности работы с массивами в языке C/C++.

8. Понятие и назначение многомерных массивов, принципы работы с ними, особенности работы с массивами в языке C/C++.

9. Понятие об указателе. Получение адреса переменных. Доступ к данным через указатель. Области применения указателей.

10. Проблемы использования указателей: утечка памяти. Приемы для снижения риска утечки памяти.

11. Проблемы использования указателей: неинициализированные указатели. Приемы для устранения риска использования неинициализированных указателей.

12. Связь массивов и указателей.

13. Динамические массивы: определение, особенности, правила работы с ними. Пример.

14. Создание функций в C++. Структура определения функции. Способы вызова функции. Пример.

15. Создание функций в C++. Формальные и фактические аргументы функции. Пример.

16. Создание функций в C++. Способы возврата результата функции. Пример.

17. Особенности передачи массивов в функцию.

18. Рекурсивная функция. Процесс рекурсивных вызовов. База рекурсивного алгоритма. Пример.

19. Понятие о строковых данных. С-строка: понятие, структура, основные принципы работы с ней. Пример.

20. Обзор основных операций над строками. Стандартная библиотека C для строковых

данных.

21. Общие сведения о файлах. Общая процедура работы с файлами.
22. Типы файлов в языке C/C++. Режимы работы с файлами.
23. Файловый ввод/вывод в стиле языка C. Пример работы с текстовым файлом.
24. Файловый ввод/вывод в стиле языка C++. Пример работы с текстовым файлом.
25. Тип структуры в C++. Определение типа структуры. Создание и инициализация объектов-структур.
26. Массивы структур.
27. Динамические объекты-структуры.
28. Абстрактные структуры данных: списки. Разновидности списков.
29. Линейные связные списки, типичные операции со списками, их реализация в C++.
30. Сравнение списков с массивами.

4.2.2. Экзамен (тестовые задания)

4.2.2.1. Порядок проведения и процедура оценивания

В случае проведения экзамена дистанционно тестовые задания загружаются в «Microsoft Teams». Экзамен в форме тестирования проводится согласно утвержденному расписанию. Тестовые задания предоставляются для каждой группы индивидуально по 30 вопросов в каждом. Задания загружаются в команду «Microsoft Teams» заранее до экзамена и запускается в день экзамена по расписанию автоматически. На решение тестовых заданий студентам дается 60 минут.

4.2.2.2. Критерии оценивания

Оценка выставляется автоматически в «Microsoft Teams», в зависимости от процента правильно выполненных заданий.

- 1) 86-100% от максимального числа баллов
От 86 до 100% правильных ответов.
- 2) 71-85% от максимального числа баллов
От 71 до 85% правильных ответов.
- 3) 56-70% от максимального числа баллов
От 56 до 70% правильных ответов.
- 4) 0-55% от максимального числа баллов
От 0 до 55% правильных ответов.

4.2.2.3. Оценочные средства

2 семестр

1. Алгоритм — это:

- правила выполнения определенных действий
- ориентированный граф, указывающий порядок исполнения некоторого набора команд
- понятное и точное предписание исполнителю совершить последовательность действий, направленных на достижение поставленных целей+
- набор команд для компьютера

2. Укажите последовательность команд, в результате выполнения, которых значения переменных X и Y поменяются местами:

- $X=X+Y; Y=X-Y; X=X-Y$
- $B=X; X=Y; Y=X$
- $X=Y; Y=X$
- $Y=X; B=X; X=Y$
- $C=X; X=Y; Y=C+$

3. Укажите синтаксически неправильную запись:

- $X=Y*\sin(X)^2+4$

- $X=Y*\sin^2(X)+4$
- $X=Y*\sin(X^2)+4$
- $X=Y^2*\sin(X)+4$

4. Укажите сколько раз выполнится цикл в программе:

$a=3; b=7;$

$\text{while } ((a / 2) \leq (b / 3))$

{

$a=a+2;$

$b=b+3;$

};

- бесконечное число раз+
- 10
- 100
- 1000
- другой ответ

5. Ветвление обязательно должно содержать

- условие и оператор, выполняемый в случае истинности условия+
- оператор, выполняемый в случае истинности условия и оператор, выполняемый в случае ложности условия
- оператор, выполняемый в случае ложности условия
- только условие

6. Расположите в правильном порядке этапы решения задач на компьютере:

- постановка задачи
- математическое описание задачи
- разработка алгоритма
- программирование
- тестовое решение задачи и анализ результатов решения
- сопровождение программы

7. Деятельность, направленная на обнаружение и исправление ошибок в программной системе называется ...

- отладкой+
- тестированием
- рефакторингом
- демонстрацией

8. В результате работы алгоритма

$Y := X + 5$

$X := Y$

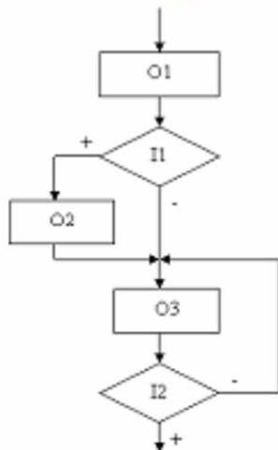
$Y := X + Y$

вывод Y

переменная Y приняла значение 14. Укажите число, которое являлось значением переменной X до начала работы алгоритма.

- 2+
- 7
- 5
- 10
- 14

Блок-схема на рисунке соответствует алгоритму



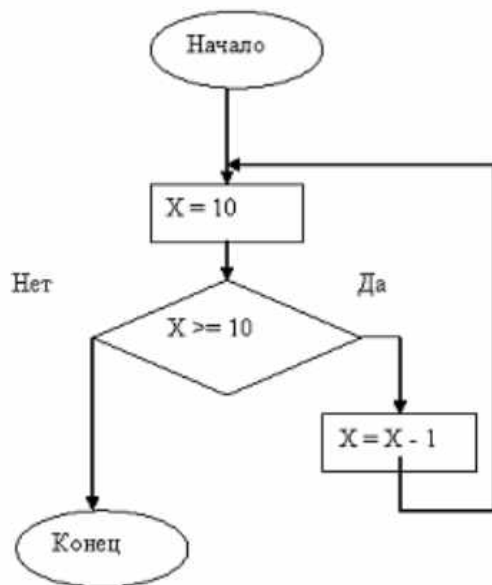
где кв – конец ветвления

кц – конец цикла

9. нц – начало цикла

- О1; Если I1 то О2; кв; О3; Пока не I2 выполнять нц О3; кц;+
- О1; Если I1 то О2; кв; Пока не I2 выполнять нц О3; кц;
- О1; Если I1 то О2; иначе О3; кв; Пока не I2 выполнять нц О3; кц;
- О1; Если I1 то О2; кв; О3; Пока I2 выполнять нц О3; кц;

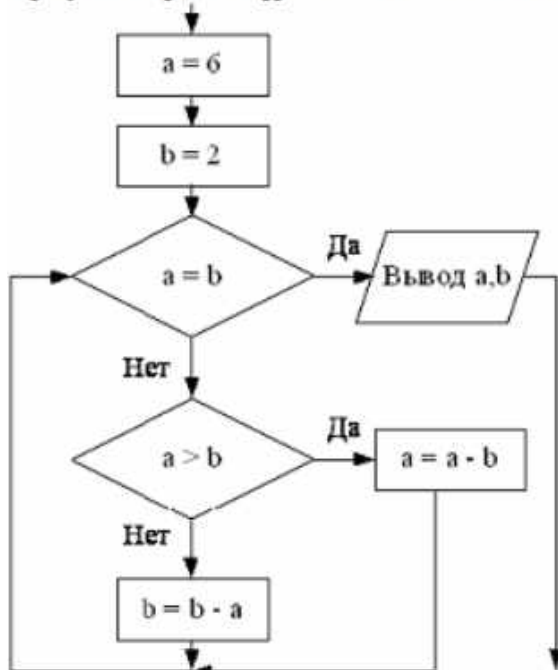
От бесконечного цикла позволяют избавиться изменения в блок-схеме, такие



10. как...

- после блока $X=X-1$ переход должен быть к блоку с условием+
- вместо действия $X=X-1$ нужно написать $X=X-10$
- вместо условия $X \geq 10$ нужно написать $X > 0$
- после действия $X=X-1$ нужно дописать еще одно действие $X=X-9$

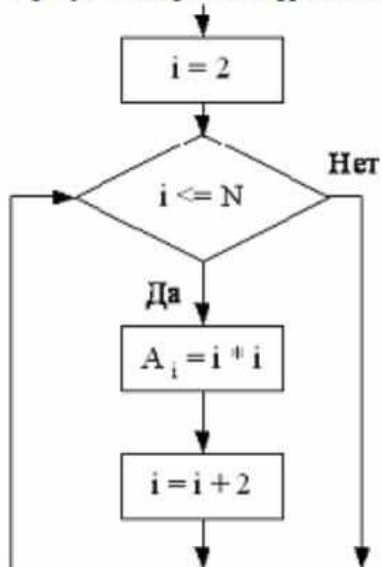
В результате работы фрагмента блок-схемы алгоритма



11. a и b примут следующие значения ...

- $a=2, b=2$
- $a=0, b=0$
- $a=2, b=4$
- $a=4, b=2$

12. В результате работы фрагмента алгоритма

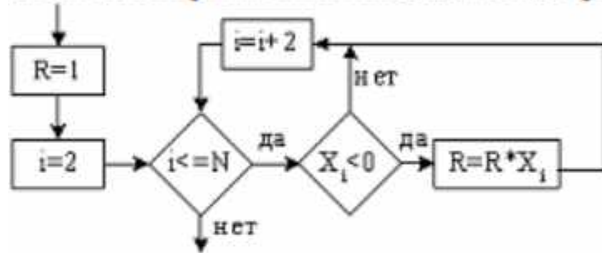


элементы массива A_2, A_4, A_6, A_8 при $N=8$ получат, соответственно, значения ...

- 4, 16, 36, 64
- 2, 4, 16, 32
- 4, 16, 32, 48
- 4, 12, 24, 36

13.

Задан одномерный массив X_1, X_2, \dots, X_N . Фрагмент алгоритма



определяет...

- произведение отрицательных элементов с четными номерами+
- количество положительных элементов с четными номерами
- произведение отрицательных элементов
- произведение положительных элементов с четными номерами

14. Утверждение «Оператор в теле цикла будет обязательно выполнен хотя бы один раз» относится к ...

- циклу с постусловием+
- циклу со счетчиком
- циклу с предусловием
- циклу с убыванием значения счетчика

15. Если задан тип данных, то известной является информация о

- диапазоне возможных значений+
- начальном значении
- количестве обращений к данным
- способе обработки

16. Рекурсия использует

- обращение подпрограммы к самой себе+
- удаление подпрограммой самой себя
- размножение подпрограммой самой себя
- заражение подпрограммой самой себя

17. Фрагмент программы:

```
char s1[] = "МИР";
char s2[] = "ТРУД\n";
fputs(s1, file);
fputs(s2, file);
fputs("МАЙ", file);
```

Каким получится содержимое файла в этом случае?

- Получим две строки. Первая строка: МИРТРУД, вторая: МАЙ+
- Получим одну строку: МИРТРУДМАЙ
- Получим одну строку: МИР ТРУД МАЙ
- Получим три строки. Первая строка: МИР, вторая: ТРУД, третья: МАЙ

18. Подпрограммам НЕ свойственно...

- усложнение понимания работы программы+
- упрощение читабельности программы
- уменьшение общего объема программы

- структурирование программы

19. Задан фрагмент алгоритма:

1. если $a < b$, то $c = b - a$, иначе $c = 2 * (a - b)$
2. $d = 0$
3. пока $c > a$ выполнить действия $d = d + 1$, $c = c - 1$

В результате выполнения данного алгоритма с начальными значениями $a = 8$, $b = 3$, переменные c и d примут значения

- $c = 8$, $d = 2$
- $c = -5$, $d = 1$
- $c = 5$, $d = 0$
- $c = 10$, $d = 1$

20. При выполнении подпрограммы

Алг пр1 (арг цел X, рез цел F)

Нач

если $X \leq 1$

то $F := 1$

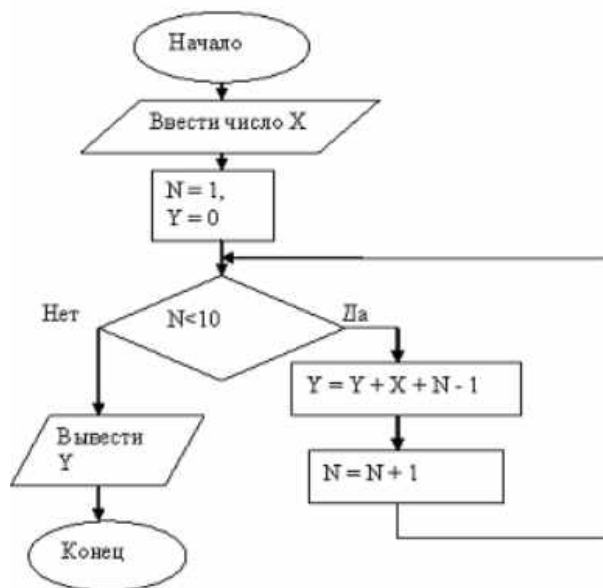
иначе $F := F(X - 2) - 4$

все кон

с параметрами (3, A) значение переменной A будет равно

- -3
- 1
- -7
- -4
- 0

21.



Данная блок-схема программы ...

- если X натуральное число, то производит сложение 9 подряд идущих натуральных чисел начиная с введенного и выводит результат+

- если X натуральное число, производит сложение 10 подряд идущих натуральных чисел начиная с введенного и выводит результат
- если X натуральное число, возводит введенное число в 10 степень и выводит результат
- если X натуральное число, возводит введенное число в 9 степень и выводит результат

22. Дан массив целых чисел $\{A_i\}$, где $i=1,2,3,\dots,M$. Пусть M равно 15. Программа вычисляет произведение сумм некоторых элементов этого массива. В программе введены следующие константы: G=1; W=12; T=8; L=15.

```
ПРОГРАММА 15;
  ФУНКЦИЯ СУММА(I1, I2);
  НАЧАТЬ ФУНКЦИЮ
  || S:=0;
  ||НЦ ДЛЯ I:=I1 ДО I2
  |||| S:=S + A[I]
  ||КЦ;
  || СУММА:=S
  КОНЕЦ ФУНКЦИИ;
НАЧАТЬ ПРОГРАММУ
|| ПИСАТЬ ('ВВЕДИТЕ ЗНАЧЕНИЯ МАССИВА А: ');
||НЦ ДЛЯ J:=1 ДО M
|||| ЧИТАТЬ (A[J]);
||КЦ;
|| P:=СУММА (G, W)*СУММА(T, L);
|| ПИСАТЬ ('ПРОИЗВЕДЕНИЕ РАВНО:', P:6)
КОНЕЦ ПРОГРАММЫ.
```

Работу программы описывает следующее алгебраическое выражение :

$$\begin{aligned}
 1. \quad P &= \sum_{i=1}^{12} A_i \cdot \sum_{j=8}^{15} A_j \\
 2. \quad P &= \sum_{i=1}^{15} A_i \cdot \sum_{j=1}^6 A_j \\
 3. \quad P &= \sum_{i=1}^8 A_i \cdot \sum_{j=1}^{12} A_j \\
 4. \quad P &= \sum_{i=1}^{12} A_i \cdot \sum_{j=8}^{12} A_j
 \end{aligned}$$

- 1+
- 2
- 3
- 4

23. Интегрированная система программирования включает компонент для перевода исходного текста программы в машинный код, который называется ...

- транслятор+
- преобразователь
- построитель кода
- переводчик

24.

```

Процедура KLMN
ПРОЦЕДУРА KLMN;
НАЧАТЬ ПРОЦЕДУРУ
|| ПИСАТЬ ('ВВЕДИТЕ ЗНАЧЕНИЕ K, L, M, N');
|| ЧИТАТЬ (K, L, M, N);
|| ЕСЛИ K=L ТО
|| | ЕСЛИ M<N ТО
|| | | X:=1
|| | | ИНАЧЕ
|| | | X:=2
|| | КОНЕЦ ЕСЛИ
|| ИНАЧЕ X:=3
|| КОНЕЦ ЕСЛИ
КОНЕЦ ПРОЦЕДУРЫ;

```

реализует следующее алгебраическое выражение ...

$$\begin{aligned}
 1. \quad X &= \begin{cases} 1, \text{ если } K = L \text{ и } M < N; \\ 2, \text{ если } K = L; \\ 3, \text{ если } K \neq L \text{ и } M \geq N. \end{cases} \\
 2. \quad X &= \begin{cases} 1, \text{ если } K = L; \\ 2, \text{ если } K \leq L \text{ и } M < N; \\ 3, \text{ если } K \neq L \text{ и } M \geq N. \end{cases} \\
 3. \quad X &= \begin{cases} 1, \text{ если } K = L; \\ 2, \text{ если } K \leq L \text{ и } M \leq N; \\ 3, \text{ если } K \neq L \text{ и } M \geq N. \end{cases} \\
 4. \quad X &= \begin{cases} 1, \text{ если } K = L \text{ и } M < N; \\ 2, \text{ если } K = L \text{ и } M \geq N; \\ 3, \text{ если } K \neq L. \end{cases}
 \end{aligned}$$

25. Исполняемый файл, содержащий программу в виде исполнимого кода, и созданный для операционной системы Windows, имеет расширение ...

- *.EXE+
- *.TXT
- *.DAT
- *.XLS

26. Параметры, указываемые в момент вызова подпрограммы из основной программы, называются ...

- Фактическими+
- глобальными
- постоянными
- абсолютными
- формальными

27. Параметры, указываемые при описании подпрограммы (в заголовке функции), называются ...

- фактическими
- глобальными
- постоянными
- абсолютными

- формальными+

28. Набор операторов, выполняющих заданное действие и не зависящих от других частей исходного кода, называют ...

- подпрограммой+
- телом программы
- параметрами программы
- разделом программы

29. Создание исполняемого файла из исходного текста программы предполагает выполнение процессов

- компиляции+
- компоновки+
- интерпретации
- исполнения программы

30. Типы входных и выходных данных определяются на этапе...

- постановки задачи+
- кодирования
- разработки алгоритма
- тестирования и отладки

31. Компилятор используется при программировании

- на языке высокого уровня+
- на языке машинных команд
- на языке сверхуровня
- на естественном языке

32. Режим интерпретации можно использовать...

- при отладке программ+
- для изменения синтаксиса языка программирования
- для изменения семантики языка программирования
- при изменении грамматики

33. Временная эффективность программного продукта зависит

- от времени выполнения программы+
- от времени, затраченного на разработку программы
- от времени, затрачиваемого на освоение методов работы с программой
- от объема используемой программой памяти

34. _____ подразумевает использование подпрограмм, каждая из которых выполняет некоторую локальную, четко формулируемую подзадачу.

- структурное проектирование
- процедурное проектирование+
- базовое проектирование
- объектно-ориентированное проектирование

35. Дана функция для вычисления координат середины отрезка, заданного на плоскости координатами своих концов:

```

void Center(double x1, double y1,
double x2, double y2,
double *xc, double *yc)
{
*xc = (x2+x1)/2;
*yc = (x2+x1)/2;
}

```

Вызов функции для нахождения середины отрезка с концами в точках (12, 10) и (0,-44) можно осуществить следующими способами:

- double Xc, Yc; Center(12, 10, 0, -44, &Xc, &Yc);+
- double *Xc, *Yc; Center(12, 10, 0, -44, Xc, Yc);+
- double *Xc, *Yc; Center(12, 10, 0, -44, *Xc, *Yc);
- double Xc, Yc; Center(12, 10, 0, -44, Xc, Yc);

36. Объем памяти для хранения значения переменной типа double (C++) -

-
- 8 байт+
 - 4 байт
 - 2 байт
 - 1 байт

37. Каким символом завершается Си-строка?

- '\0'+
- '.'
- 'n'
- '\n'

38. Какой стандартный код используется для символьных данных типа char в C++?

- Код ASCII
- Код ASCII+
- Код UTF-8
- Код cp-1251

39. Дан фрагмент программы

```

.....
doubleAvgVal( double v1, double v2 )
{
return (v1 + v2) / 2;
}
void main()
{
.....
doubleres, n1, n2;
printf("Введите два числа через пробел...\n");
scanf("%lf %lf", &n1, &n2);
res = AvgVal( n1, n2 );
printf("Для чисел %0.1f и 0.1f результат равен %0.1f.\n", n1, n2, res);
.....
}

```

Укажите переменные, которые являются фактическими параметрами функции AvgVal.

- n1+
- n2+
- v1
- v2
- res

40. Дан фрагмент программы

```
.....  
doubleAvgVal( double v1, double v2 )  
{  
return (v1 + v2) / 2;  
}  
void main()  
{  
.....  
double res, n1, n2;  
printf("Введите два числа через пробел...\n");  
scanf("%lf %lf", &n1, &n2);  
res = AvgVal( n1, n2 );  
printf("Для чисел %0.1f и %0.1f результат равен %0.1f.\n", n1, n2, res);  
.....  
}
```

Укажите переменные, которые являются формальными параметрами функции AvgVal.

- n1
- n2
- v1+
- v2+
- res

41. Укажите способы возврата из функции данных (в C++), полученных при ее выполнении

- через имя функции+
- через формальные параметры типа указателя+
- через формальные параметры типа типа ссылки+
- через формальные параметры типа типа int
- через формальные параметры типа типа double
- данные из функции вернуть нельзя

42. Дан фрагмент программы

```
doubleMinVal( double v1, double v2)  
{  
return (v1 < v2 ? v1 : v2);  
}  
doubleMaxVal( double v1, double v2)  
{  
return (v1 > v2 ? v1 : v2);  
}
```

```

void main()
{
double x1(25.0), x2(29.0), x3(35.0);
double minv, maxv, res;
minv = MinVal( MinVal(x1, x2), x3);
maxv = MaxVal( MaxVal(x1, x2), x3);
res =minv +maxv ;
.....
}

```

Какое значение примет переменная res после выполнения программы?

Правильный ответ - 60

43. Какие из указанных утверждений правильные

- массив в программе на C++ неявно представляет собой указатель на начальный элемент+
- в массиве не хранится никакой информации о его размере+
- при передаче одномерного массива в функцию используют один формальный аргумент типа указателя на начальный элемент массива и формальный аргумент для передачи размера массива+
- передать в функцию можно только одномерный массив

44. Глобальные переменные в программе - это ...

- переменные, которые объявляют в начале программы, до начала определения функций+
- переменные, которые указывают при вызове функции
- переменные, определяемые в теле любой функции (в том числе и главной)

45. Локальные переменные в программе - это ...

- переменные, которые объявляют в начале программы, до начала определения функций
- переменные, которые указывают при вызове функции
- переменные, определяемые в теле любой функции (в том числе и главной)+

46. Какую функцию используют для ввода с клавиатуры C-строки

- gets+
- puts
- strcpy
- strcat

47. Для осуществления доступа к адресуемому значению указателя применяют операцию

- разыменования+
- разадресации
- развязывания
- распрямления

48. Динамические массивы используются для

- создания массивов ровно такого размера, какой нужен в данный момент+
- создания многомерных массивов
- создания массивов с большим числом элементов

49. Пусть имеется некоторая переменная A типа T и указатель pA типа T* на переменную A. Тогда значение переменной A можно получить, указав ...

- A+
- *pA+
- &pA
- *A
- &A
- pA

50. Какая из следующих функций добавляет одну строку в конец другой?

- Strcat+
- strncpy
- puts
- strcmp

Перечень литературы, необходимой для освоения дисциплины (модуля)

Направление подготовки: 01.03.02 - Прикладная математика и информатика

Профиль подготовки: отсутствует

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2021

Литература:

1. Конова Е. А. Алгоритмы и программы. Язык С++ : учебное пособие / Е. А. Конова, Г. А. Поллак. - 5-е изд., стер. - Санкт-Петербург : Лань, 2020. - 384 с. - ISBN 978-5-8114-5431-0. - URL : <https://e.lanbook.com/book/140730> (дата обращения: 05.08.2020). - Текст : электронный.
2. Окулов С. М. Основы программирования : учебное пособие / С. М. Окулов. - 10-е изд. - Москва : Лаборатория знаний, 2020. - 339 с. - ISBN 978-5-00101-759-2. - URL : <https://e.lanbook.com/book/135560> (дата обращения: 06.08.2020). - Текст : электронный.
3. Дорогов В.Г. Основы программирования на языке С : учебное пособие / В.Г. Дорогов, Е.Г. Дорогова ; под ред. проф. Л.Г. Гагариной. - Москва : ФОРУМ : ИНФРА-М, 2019. - 224 с. - ISBN 978-5-8199-0882-2. - URL : <http://znanium.com/catalog/product/1016471> (дата обращения: 14.07.2020). - Текст : электронный.
4. Колдаев В.Д. Структуры и алгоритмы обработки данных : учебное пособие / В. Д. Колдаев. - Москва: ИЦ РИОР, 2014. - 296 с. - (Высшее образование:Бакалавриат). - ISBN 978-5-369-01264-2. - URL : <http://znanium.com/go.php?id=418290> (дата обращения: 14.07.2020). - Текст : электронный.
5. Серебряков В.А. Теория и реализация языков программирования : учебное пособие / В. А. Серебряков. - Москва : ФИЗМАТЛИТ, 2012. - 236 с. - ISBN 978-5-9221-1417-2. - URL: <https://e.lanbook.com/book/5294> (дата обращения: 14.07.2020). - Текст : электронный.
6. Шлее М. Qt4.5. Профессиональное программирование на С++ : практическое руководство / М. Шлее - Санкт-Петербург: БХВ-Петербург, 2010. - 882 с. - ISBN 978-5-9775-0398-3 - URL: <http://znanium.com/catalog/product/350671> (дата обращения: 14.07.2020). - Текст : электронный.
7. Златопольский Д. М. Программирование: типовые задачи, алгоритмы, методы / Д. М. Златопольский. - 4-е изд. - Москва : Лаборатория знаний, 2020. - 226 с. - ISBN 978-5-00101-789-9. - URL : <https://e.lanbook.com/book/135562> (дата обращения: 06.08.2020). - Текст : электронный.
8. Немцова Т.И. Программирование на языке высокого уровня. Программирование на языке С++ : учебное пособие / Т.И. Немцова, С.Ю. Голова, А.И. Терентьев ; под ред. Л.Г. Гагариной. - Москва : ФОРУМ : ИНФРА-М, 2019. - 512 с. - (Среднее профессиональное образование). - ISBN 978-5-8199-0699-6. - URL : <http://znanium.com/catalog/product/1000008> (дата обращения: 14.07.2020). - Текст : электронный.

Перечень информационных технологий, используемых для освоения дисциплины (модуля), включая перечень программного обеспечения и информационных справочных систем

Направление подготовки: 01.03.02 - Прикладная математика и информатика

Профиль подготовки: отсутствует

Квалификация выпускника: бакалавр

Форма обучения: очная

Язык обучения: русский

Год начала обучения по образовательной программе: 2021

Освоение дисциплины (модуля) предполагает использование следующего программного обеспечения и информационно-справочных систем:

Операционная система Microsoft Windows 7

Пакет офисного программного обеспечения Microsoft Office

Браузер MozillaFirefox

Adobe Acrobat Reader

Антивирус Касперского

Qt Creator

Microsoft Visual Studio

ЭБС "ZNANIUM.COM"

ЭБС Издательства "Лань"

ЭБС "Консультант студента"