

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новгородский государственный университет имени Ярослава Мудрого»
Центр подготовки кадров в сфере информационных технологий:
кампус «Школа 21»



УТВЕРЖДАЮ

Проректор по ОД

Ю.В. Данейкин

« 04 » февраля 20 25 г.

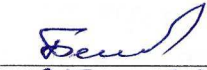
**ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА –
ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ**

ВВЕДЕНИЕ В ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C

Лицензия: рег. № Л035-00115-53/00119498 от 13.04.2016,
выданная Рособрнадзором на срок – бессрочно

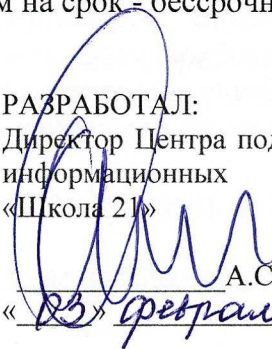
СОГЛАСОВАНО:

Начальник ОДОП

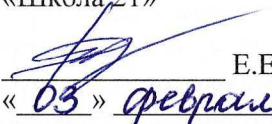

Е.И. Белова
« 05 » 02 2025 г.

РАЗРАБОТАЛ:

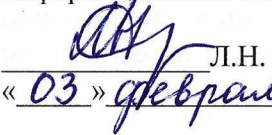
Директор Центра подготовки кадров в сфере
информационных технологий: кампус
«Школа 21»


А.С. Латышев
« 03 » февраля 2025 г.

Зам. директора Центра подготовки кадров в
сфере информационных технологий: кампус
«Школа 21»


Е.Е. Деева
« 03 » февраля 2025 г.

Старший преподаватель кафедры
информационных технологий и систем


Л.Н. Цымбалюк
« 03 » февраля 2025 г.

1. ХАРАКТЕРИСТИКА ДОПОЛНИТЕЛЬНОЙ ПРОФЕССИОНАЛЬНОЙ ПРОГРАММЫ

Общие сведения дополнительной профессиональной программы повышения квалификации (далее – программа) «Введение в программирование на языке C»

Предшествующий уровень образования слушателя	–	среднее профессиональное, высшее образование
Срок освоения (продолжительность обучения)	–	144 часа
Форма обучения	–	очно-заочная с использованием ДОТ
Форма итоговой аттестации	–	зачет
Дополнительные сведения	–	программа предназначена для специалистов, использующих язык программирования C в профессиональной деятельности

Цель программы: совершенствование и (или) получение новых компетенций, необходимых для выполнения профессиональной деятельности, и (или) повышение профессионального уровня в рамках имеющейся квалификации работника (слушателя) по использованию языков и методов программирования, формированию научного, творческого подхода к освоению технологий, методов и средств производства программного обеспечения, овладение знаниями в области технологии программирования.

Задачи программы:

- Дать определение основным типам данных языка;
- Определить управляющие последовательности, используемые в языке;
- Определить сложные типы данных языка C;
- Описать основные возможности по файловому вводу-выводу.

Планируемые результаты обучения направлены на совершенствование и актуализацию необходимых в деятельности слушателей следующих компетенций:

- Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности;
- Способен разрабатывать алгоритмы и программы, пригодные для практического применения;
- Способен осваивать методики использования программных средств для решения практических задач.

Учет в содержании программы профессиональных стандартов:

- в программе учитывается профессиональный стандарт 06.001 Программист (Утвержден приказом Министерства труда и социальной защиты Российской Федерации от 20.07.2022 № 424н) для следующих трудовых функций: C/01.5 Разработка процедур интеграции программных модулей, C/02.5 Осуществление интеграции программных модулей и компонентов и проверки работоспособности выпусков программного продукта.

Учет в содержании программы квалификационных требований, указанных в квалификационных справочниках по соответствующим должностям, профессиям и специальностям, профессиям и специальностям:

-в программе учитываются квалификационные требования, указанные в Едином квалификационном справочнике должностей руководителей, специалистов и служащих, раздел "Общепрофессиональные квалификационные характеристики должностей работников, занятых на предприятиях, в учреждениях и организациях" для должностей: Инженер-программист (программист).

Квалификационные требования утверждены Постановлением Минтруда России от 21.08.1998 N 37 (Редакция от 9 апреля 2018 года (в т.ч. с изменениями вступ. в силу 01.07.2018) "Об утверждении Единого квалификационного справочника должностей руководителей, специалистов и служащих, раздел "Общепрофессиональные квалификационные характеристики должностей работников, занятых на предприятиях, в учреждениях и организациях".

Программа разработана с учетом следующих нормативных документов:

1. Федеральный закон "Об образовании в Российской Федерации" от 29.12.2012 N 273-ФЗ

2. Порядок организации и осуществления образовательной деятельности по дополнительным профессиональным программам (утв. приказом Министерства образования и науки РФ от 1 июля 2013 г. N 499)

3. Письмо Минобрнауки России от 22.04.2015 N ВК-1032/06 "О направлении методических рекомендаций" (вместе с "Методическими рекомендациями разъяснениями по разработке дополнительных профессиональных программ на основе профессиональных стандартов")

4. Положение о порядке разработки дополнительных образовательных программ в федеральном государственном бюджетном образовательном учреждении высшего образования «Новгородский государственный университет имени Ярослава Мудрого» от 27.04.2021

4. Образовательный стандарт ФГОС ВО 09.03.01 Информатика и вычислительная техника (Приказ Минобрнауки России от 19 сентября 2017 г. № 929 Об утверждении федерального государственного образовательного стандарта высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника»).

В результате успешного освоения программы слушатель получает удостоверение о повышении квалификации установленного образца.

2. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ

Процесс освоения программы направлен на формирование следующих компетенций:

- способен использовать языки, утилиты и среду программирования, средства пакетного выполнения процедур в профессиональной деятельности;
- способен производить настройки параметров программного продукта и осуществлять запуск процедур сборки;
- способен проектировать компьютерное программное обеспечение.

В результате освоения программы «Введение в программирование на языке С» слушатель должен:

знать:

- методы и средства сборки модулей и компонентов компьютерного программного обеспечения;
- интерфейсы взаимодействия с внешней средой;
- интерфейсы взаимодействия внутренних модулей системы
- методы и средства разработки процедур для развертывания компьютерного программного обеспечения
- языки, утилиты и среды программирования, средства пакетного выполнения процедур
- основные стандарты оформления технической документации на компьютерное программное обеспечение;
- методы и средства сборки и интеграции программных модулей и компонентов;
- интерфейсы взаимодействия программного продукта с внешней средой;
- интерфейсы взаимодействия внутренних модулей программного продукта;
- методы и средства проверки работоспособности выпусков программных продуктов;
- языки, утилиты и среды программирования, средства пакетного выполнения процедур;

уметь:

- писать программный код процедур интеграции программных модулей;
- выполнять процедуры сборки программных модулей и компонентов в программный продукт;
- производить настройки параметров программного продукта и осуществлять запуск процедур сборки;
- проводить проверку работоспособности программного продукта;
- использовать командные средства разработки компьютерного программного обеспечения;
- осуществлять коммуникации с заинтересованными сторонами;

владеть:

- разработкой и документированием программных интерфейсов;
- разработкой процедур сборки модулей и компонентов компьютерного программного обеспечения;
- разработкой процедур миграции и преобразования (конвертации) данных;
- оценкой и согласованием сроков выполнения поставленных задач;
- сборкой программных модулей и компонентов в программный продукт;
- подключением программного продукта к компонентам внешней среды;
- проверкой работоспособности выпусков программного продукта.

3. УЧЕБНЫЙ ПЛАН

Учебный план программы «Введение в программирование на языке С»

№ п/п	Наименование разделов, тем	Всего часов	В том числе:			Формы контроля *
			лекции	практические и лабораторные занятия	самостоя- тельная работа	
1	Подготовка к работе	20	6	6	8	
1.1	Знакомство с Linux	8	2	2	4	
1.2	Система контроля версий Git	12	4	4	4	
2	Введение в язык С	122	22	36	64	
2.1	Введение в структурное программирование	10	2	4	4	
2.2	Указатели, массивы, матрицы и работа с памятью	26	4	8	14	
2.3	Технологии программирования на С	16	4	4	8	
2.4	Строки и пользовательские структуры данных	16	2	4	10	
2.5	Динамические структуры данных	28	6	8	14	
2.6	Работа с файлами	26	4	8	14	
	Итоговая аттестация	2		2		зачет
	Итого	144	28	44	72	

* - промежуточная аттестации и текущий контроль в программе не предусмотрены

4. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Календарный учебный график* программы «Введение в программирование на языке С»

Программа реализуется в течение 4 недель (с понедельника по субботу), по 6 часов (3 аудиторных + 3 СРС) занятий в день (с 9 до 11.45 без перерыва на обед). Расписание занятий составляется на основании данной информации.

5. РАБОЧАЯ ПРОГРАММА УЧЕБНЫХ ПРЕДМЕТОВ, КУРСОВ, ДИСЦИПЛИН (МОДУЛЕЙ), РАЗДЕЛОВ, ТЕМ

Темы и содержание лекций

Тема 1.1 Знакомство с Linux	Серверные дистрибутивы. Загрузка Linux. Командная оболочка. Протокол Secure SHell. Установка программ (утилит) пакетным менеджером. Структура файловой системы и работа с файлами. Права доступа. Команды для работы с операционной системой в консоли.
Тема 1.2. Система контроля версий Git	Система контроля в контексте Git. Внутреннее строение Git. Репозитории. Коммиты, пуши. Ветви и мерджинг. Пул-реквесты.
Тема 2.1. Введение в структурное программирование	Арифметические и логические выражения. Префиксный и постфиксный инкремент и декремент. Приоритеты. Основные конструкции языка структурированного программирования: последовательность операторов, ветвление, цикл. Циклы while, do-while, for. Тело цикла, итерация, вложенные циклы.
Тема 2.2. Указатели, массивы, матрицы и работа с памятью	Типы данных: целочисленный и с плавающей запятой, знаковые и беззнаковые. Точность значения с плавающей запятой. Диапазоны значений. Приведение типов: явное и неявное. Прямая и косвенная адресация, указатели. Операторы взятия адреса и разыменования. Массив как агрегатный тип данных. Объявление и инициализация массивов. Индексирование элементов. Размещение в памяти. Заполнение случайными данными. Передача массивов в функцию. Многомерные массивы: объявление и работа с ними. Размещение многомерных массивов в памяти, передача в функцию. Память как последовательность одинаковых пронумерованных ячеек. Байты, биты. Переменные: объявление, инициализация, правила именования.
Тема 2.3. Технологии программирования на C	Работа в интегрированной среде разработки (IDE). Создание, компилирование, запуск программы. Базовые методы отладки. Пример простейшей программы. Операторы. Переменные. Константы: литеральные и символические. Присваивание. Выражения. Общая структура программ на языке C. Главная функция main. Использование функций: декомпозиция задач, библиотеки кода. Проектирование сверху-вниз и снизу-вверх. Объявление, определение и вызов функции. Аргументы и параметры функции. Возвращаемое значение, void-функции. Локальные переменные, область видимости. Оператор return. Передача параметров по указателю.
Тема 2.4. Строки и пользовательские структуры данных	Символ, кодировка, ASCII. Управляющие символы. Работа с символами. Символ как числовая константа. Ввод и вывод символов. Escape-последовательности. Строки как массивы символов. Маркер конца строки. Массивы в качестве глобальных переменных. Объявление и инициализация строк, чтение с клавиатуры и из файла. Передача строк в функцию. Библиотечные функции для работы со строками.
Тема 2.5. Динамические структуры данных	Характеристика динамических структур данных. Порядок

	работы с динамическими структурами данных. Достоинства и недостатки связного представления данных. Классификация динамических структур данных: однонаправленные (односвязные) списки; двунаправленные (двусвязные) списки; циклические списки; стек; дек; очередь; бинарные деревья. Объявление динамических структур данных. Поле данных. Доступ к данным в динамических структурах. Работа с памятью при использовании динамических структур.
Тема 2.6. Работа с файлами	Консоль. Буфер ввода. Форматированный ввод/вывод, форматная строка, спецификаторы преобразования, ширина и точность ввода/вывода. Файловый ввод/ вывод. Открытие и закрытие файлов. Режимы работы с файлами: чтение, запись, дозапись. Тип FILE и макроконстанта EOF.

Содержание практических занятий

№ темы	Содержание практического занятия	Объем, час
1.1	Знакомство с Linux Задание практической работы: Откройте терминал. Ознакомьтесь с возможностями команды pwd с помощью команды man: Определите текущий каталог, в котором вы находитесь командой pwd: Ознакомьтесь с возможностями команды cd с помощью команды man: Перейдите в корневой каталог командой cd Ознакомьтесь с возможностями команды ls с помощью команды man: Просмотрите содержимое корневого каталога командой ls: Сделайте копию экрана для использования в отчете по лабораторной работе . Вернитесь в домашний каталог, используя команду cd без параметров: Ознакомьтесь с возможностями команды mkdir с помощью команды man: Создайте каталог «test», используя команду mkdir: Перейдите в каталог «test», используя команду cd: Просмотрите содержимое каталога, используя команду ls: Создайте каталог «test2», используя команду mkdir: Ознакомьтесь с возможностями команды touch с помощью команды man: Создайте файл «text» в каталоге «test2» используя команду touch: Ознакомьтесь с возможностями команды mv с помощью команды man: Переименуйте файл «text» в «textSIT» используя команду mv Ознакомьтесь с возможностями команды cp с помощью команды man: Скопируйте файл «textSIT» в каталог «test2» под именем «copy.txt», используя команду cp: Ознакомьтесь с возможностями команды ln с помощью команды man: Создайте жесткую ссылку «link» на файл «copy.txt» используя команду ln: Создайте символическую ссылку «simlink» на файл «copy.txt» используя команду ln: Просмотрите результаты в текущем каталоге при помощи команды ls с аргументами la: Сделайте копию экрана для использования в отчете по лабораторной работе . Удалите созданные вами файлы и ссылки в лабораторной работе используя команду rm Сделайте копию экрана для использования в отчете по лабораторной работе . <i>Вопросы к практической работе:</i> Чем отличается вывод команд ls -F и ls -la? С помощью какой команды и как можно переместить файл в другой каталог? Какие действия вы совершаете, нажимая на кнопки «стрелка вверх» и «стрелка вниз»? Куда вы переходите, выполнив команду cd без параметров? Как посмотреть, какие еще параметры можно задать команде ls? Что такое «жесткая ссылка»? Что такое «символическая ссылка»?	2

	Как осуществить просмотр подкаталогов и их содержимого ? Как осуществить вывод содержимого каталога с запятыми в качестве разделителя? Как осуществить просмотр скрытых файлов в домашнем каталоге? Как осуществить создание нового каталога и необходимых подкаталогов рекурсивно? Как осуществить рекурсивное копирование всех файлов из одного каталога в другой? Как осуществить рекурсивное копирование всех файлов и подкаталогов из одного каталога в другой? Как рекурсивно удалить все файлы и подкаталоги в определенном каталоге?																
1.2	Система контроля версий Git Выполните приведенные ниже задания. 1. Изучить теоретическую часть работы. 2. Зайти в папку T://{Номер группы} и в ней создать папку соответствующую инициалам студента на английском языке. Например, для студента Иванов Петр Петрович, папка будет иметь имя IPP. 3. Провести инициализацию репозитория в созданной папке. Для этого, открыть программу Git Bash, перейти в созданную папку (для перемещения используется команда <code>cd T://{Номер группы}/{Инициалы}</code>). 4. Установить настройки имени и e-mail'a, не используя опцию <code>--global</code>. 5. Создать в папке файл <code>my_first_file.txt</code> и проиндексировать его. 6. Сделать первый коммит. 7. Открыть файл <code>my_first_file.txt</code> и добавить в него строку "test row". Проиндексировать изменения. 8. Создать новый файл <code>my_second_file.txt</code>. Проиндексировать изменения. 9. Сделать второй коммит.	4															
2.1	Введение в структурное программирование Выполните приведенные ниже задания. 1. Даны три целых числа: A, B, C. Напишите выражение, проверяющее истинность высказывания: «Справедливо двойное неравенство $A < B < C$». 3.8 Даны два целых числа: A и B. Напишите выражение, проверяющее истинность высказывания: «Числа A и B имеют одинаковую четность». 2. Напишите выражение, проверяющее, что среди чисел A, B, C и D есть как минимум 2 нечетных. 3. Дано трехзначное число. Напишите выражение, проверяющее истинность высказывания: «Все цифры данного числа различны». 4. Напишите выражение, проверяющее, что $(A + B)$ равно удвоенному остатку от деления $(C + D)$ на B. 5. Напишите выражение, проверяющее, что A равно B, без использования оператора <code>==</code>. 6. Напишите выражение, проверяющее, что A меньше B, без использования оператора <code><</code>. Y) имеет следующую таблицу истинности: <table border="1"> <tr><td>x</td><td>y</td><td>x -> y</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </table> То есть из лжи может следовать все что угодно, а из правды – только правда. Эта логическая операция называется импликацией. 7. Даны координаты двух различных полей шахматной доски x_1, y_1, x_2, y_2 (целые числа, лежащие в диапазоне 1–8). Напишите выражение, проверяющее истинность высказывания: «Ладья за один ход может перейти с поля (x_1, y_1) на поле (x_2, y_2)». 8. Напишите выражение, проверяющее, что из того, что A равняется B, следует, что B равняется A. 9. Даны два целых числа A и B ($A < B$). Вывести в порядке убывания все целые числа, расположенные между A и B (не включая числа A и B), а также количество напечатанных чисел. 10. Даны два целых числа A и B ($A < B$). Найти произведение всех целых чисел от A до B включительно. 11. Дано целое число N (> 1). Найти наименьшее целое число K, при котором выполняется неравенство $3K > N$. 12. Палиндромом называется число или фраза текста, которая читается одинаково как слева направо, так и справа налево. 13. Основы структурированного программирования Например, каждое из следующих пятизначных целых чисел является палиндромом: 12321, 55555, 45554 и 11611. Напишите программу, которая считывает с клавиатуры пятизначное целое число и определяет, является ли оно палиндромом. 14. Спросить у пользователя число N и вывести квадрат со стороной размера N	x	y	x -> y	0	0	1	0	1	0	1	0	0	1	1	1	4
x	y	x -> y															
0	0	1															
0	1	0															
1	0	0															
1	1	1															

	следующим образом. Контур квадрата выводится символом #, а внутренности символом +. Например, для $N = 4$ нужно вывести: ##### #++# #++# ##### а для $N = 2$: ## ## 15. Подсчитать сумму всех нечетных чисел от 1 до 99. 16. Подсчитать количество цифр вводимого с клавиатуры целого неотрицательного числа. 17. Шестизначный номер билета называют «счастливым», если в его записи сумма первых трех цифр равна сумме последних трех. Программа спрашивает у пользователя шестизначный номер билета и проверяет, является ли он счастливым. Если билет счастливый, то программа выводит на новой строке «Ваш билет счастливый!», иначе выводится сообщение «Ваш билет не является счастливым.»	
2.2	Указатели, массивы, матрицы и работа с памятью Выполните приведенные ниже задания. 1. Напишите программу, которая спрашивает у пользователя число N. Если $N < 1$ или $N > 100$, то скажите пользователю, что он ввел неверное число, и спросите заново. Так делайте до тех пор, пока пользователь не введет число из заданного диапазона. Затем ваша программа должна создать файл input.txt, в который на первую строку она запишет число N, а на вторую строку – N целых случайных чисел, разделенных пробелами. Этот файл можно будет использовать как входной файл для дальнейших упражнений. 2. Создайте массив заведомо большого размера (например, из 1000 элементов), затем считайте в него все числа из файла input.txt. После этого найдите количества чисел в этом массиве, которые меньше последнего элемента, больше последнего элемента и равны последнему элементу последовательности. Сохраните в выходной файл сам массив, а также найденные три числа. Информацию, выводимую в файл, продублируйте на экран. 3. Напишите программу, которая считывает все числа из файла input.txt в массив заведомо большего размера, спрашивает у пользователя произвольное число и вставляет его в массив. Для этого она спрашивает сначала число a – то, которое надо вставить, а затем число i – номер той ячейки, в которую его надо вставить. При этом программа должна спрашивать у пользователя число i до тех пор, пока оно не будет принадлежать отрезку $[0, N]$. Затем она должна вставить число a в ячейку i, сдвинув при этом подмассив $(a_i, \dots, a_{N-1})$ на одну ячейку вправо. Например, из файла мы считали массив $(1, 4, 2, 10, 3)$. Затем пользователь ввел $a = 7, i = 3$. Мы должны получить массив $(1, 4, 2, 7, 10, 3)$. Выведите результат на экран и в файл. 4. Модифицируйте программу из 7.9 так, чтобы она удаляла элемент из массива. Программа должна спрашивать только число i, следя, чтобы оно было из нужного диапазона. Затем она должна удалить i-тое число из массива, сдвинув подмассив $(a_{i+1}, \dots, a_{N-1})$ на одну ячейку влево. Например, из файла мы считали массив $(1, 4, 2, 10, 3)$. Затем пользователь ввел $i = 1$. Мы должны получить массив $(1, 2, 10, 3)$. Выведите результат на экран и в файл. 5. Модифицируйте программу из 7.10 так, чтобы она удаляла из массива не один элемент, а целый блок элементов. Программа должна спрашивать два числа i и j, следя при этом, чтобы $0 \leq i \leq j \leq N - 1$. Затем она должна удалить из массива все числа, начиная с i-го и заканчивая j-тым, сдвинув подмассив $(a_{j+1}, \dots, a_{N-1})$ на нужное количество ячеек влево. Например, из файла мы считали массив $(1, 4, 2, 10, 3)$. Затем пользователь ввел $i = 1, j = 3$. Мы должны получить массив $(1, 3)$. Выведите результат на экран и в файл. 6. Напишите программу, которая считает из файла в массив числа следующим образом: если очередное считываемое из файла число уже было считано до этого (т.е. уже есть в массиве), то оно игнорируется, в противном случае оно записывается в массив. Выведите исходный и получившийся массивы на экран. 7. Решето Эратосфена – простой алгоритм нахождения всех простых чисел до некоторого целого числа n, созданный древнегреческим математиком Эратосфеном. Рассмотрим пример для $n = 20$. Запишем натуральные числа от 2 до 20, в ряд: 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 Первое число в списке 2 – простое. Пройдем по ряду чисел, вычеркивая все числа, кратные 2 (кроме самой двойки): 2 3 5 7 9 11 13 15 17 19 Следующее невычеркнутое число 3 – простое. Пройдем по ряду чисел, вычеркивая все числа, кратные 3 (кроме самой тройки): 2 3 5 7 11 13 17 19 И так далее пока квадрат очередного невычеркнутого числа меньше n. Напишите программу, которая спрашивает у пользователя число N, заполняет заранее созданный массив N натуральными числами, а затем использует вышеизложенный	8

	алгоритм для нахождения всех простых чисел на отрезке $[2, N]$, выводя при этом все шаги работы данного алгоритма. Т.е. сначала надо вывести весь массив, затем вывести содержимое массива, когда из него удалены числа, кратные 2, и т.д. Операцию вычеркивания можно заменить обнулением соответствующих элементов массива. 8. Дана квадратная матрица A размерности $M \times M$. Найти сумму элементов ее главной диагонали, то есть диагонали, содержащей следующие элементы: $A_{0,0}, A_{1,1}, A_{2,2}, \dots, A_{M-1,M-1}$. 9. В матрице A поменять местами наибольший и наименьший элементы (можно считать, что такие элементы в матрице уникальны). Полученную матрицу вывести на экран. 10. В каждой строке матрицы A, размером $n \times m$ программа находит наименьший элемент, затем среди этих чисел выбирает наибольшее. Нужно напечатать индексы элемента с найденным значением. 11. Дана квадратная матрица A порядка M. Начиная с элемента $A_{0,0}$, вывести ее элементы следующим образом («уголками»): все элементы первой строки; элементы последнего столбца, кроме первого (уже выведенного) элемента; оставшиеся элементы второй строки; оставшиеся элементы предпоследнего столбца и т.д. Последним выводится элемент $A_{M-1,0}$. 12. Дана квадратная матрица A порядка M (M – нечетное число). Начиная с элемента $A_{0,0}$ и перемещаясь по часовой стрелке, вывести все ее элементы по спирали: первая строка, последний столбец, последняя строка в обратном порядке, первый столбец в обратном порядке, оставшиеся элементы второй строки и т. д.; последним выводится центральный элемент матрицы. 13. Дана целочисленная матрица размера $M \times N$. Найти количество ее строк, все элементы которых различны. 14. Дана матрица размера $M \times N$. Элемент матрицы называется ее локальным минимумом, если он меньше всех окружающих его элементов. Заменить все локальные минимумы данной матрицы на нули. 15. Напишите программу, которая выводит на экран фигуру в виде ромба. Вы можете использовать операторы <code>printf</code>, которые выводят либо одну звездочку, либо один пробел. Высота ромба вводится отдельно с клавиатуры. 16. Начальный вклад в банке равен 1000 руб. Через каждый месяц размер вклада увеличивается на P процентов от имеющейся суммы (P – вещественное число, $0 < P < 25$). По данному P определить, через сколько месяцев размер вклада превысит 1100 руб., и вывести найденное количество месяцев K (целое число) и итоговый размер вклада S (вещественное число). 17. Целое число называется совершенным, если сумма его делителей, включая 1 (но не само число), равна этому числу. Например, 6 является совершенным числом, поскольку $6 = 1 + 2 + 3$. Напишите программу, которая определяет, является ли введенное пользователем число совершенным. 18. Распечатать все совершенные числа в диапазоне от 1 до 1000. Напечатайте все делители для каждого совершенного числа. 19. Найти наибольшее и наименьшее значения функции $y = 3x^2 + x - 4$ на заданном интервале $[a, b]$. x изменяется с шагом 0.1. 4.11 В 1202 году Итальянский математик Леонард Пизанский (Фибоначчи) предложил такую задачу: пара кроликов каждый месяц дает приплод – двух кроликов (самца и самку), от которых через два месяца уже получается новый приплод. Согласно условию задачи, числа, соответствующие количеству пар кроликов, которые появляются через каждый месяц, составляют последовательность 1, 1, 2, 3, 5, 8, 13, 21, ..., в которой каждый элемент последовательности равен сумме двух предыдущих: $u_{n+2} = u_{n+1} + u_n$. Эту последовательность называли числами Фибоначчи. Составьте программу, выводящую все числа Фибоначчи, меньшие заданного числа N. 20. Известно, что сумма N первых нечетных чисел равна квадрату числа N, например, $1 + 3 + 5 = 3^2$, $1 + 3 + 5 + 7 = 4^2$ и т.д. Ввести натуральное K и распечатать таблицу всех натуральных чисел от 1 до K и их квадратов, вычисленных с помощью указанного соотношения (использовать операцию умножения нельзя).	
2.3	Технологии программирования на C Выполните приведенные ниже задания. 1 Написать функцию <code>void swap(int *a, int *b)</code>, меняющую значения переменных <code>a</code> и <code>b</code>. Продемонстрировать ее работу на двух числах, введенных с клавиатуры. 2 Написать функцию <code>double circle_length(double radius)</code>. С помощью этой функции	4

	написать программу, запрашивающую у пользователя радиус и вычисляющую длину окружности. 3 Написать функцию <code>int factorial(int n)</code>, вычисляющую факториал переданного целого числа. Используя данную функцию вывести факториалы чисел от 1 до 10. 4 Написать функцию <code>int is_prime(int n)</code>, проверяющую, что переданное ей целое число – простое. Вывести все простые числа от 1 до 100. 5. Написать функцию <code>double hypotenuse (double a, double b)</code>, которая вычисляет длину гипотенузы прямоугольного треугольника по двум другим сторонам. Используйте эту функцию в программе для определения длины гипотенузы треугольников, стороны которых пользователь вводит с клавиатуры. 6. Написать функцию <code>double power(double x, int n)</code>, возводящую число x в степень n. Использовать эту функцию в программе, возводящей в степень действительные числа. Число x и показатель степени n вводятся с клавиатуры. 7. Написать функцию <code>double minVal(double a, double b, double c)</code>, которая возвращает наименьшее из трех чисел с плавающей точкой. 8. Написать функцию <code>void mark(int val, int *markVal)</code>, которая принимает на вход балл в диапазоне от 0 до 100 и возвращает через параметр-указатель <code>markVal</code> оценку, соответствующую этому баллу согласно следующим правилам: 0-49 – 2, 50-69 – 3, 70-84 – 4, 85-100 – 5. Реализуйте программу, которая просит ввести балл с клавиатуры, а затем выводит соответствующую ему оценку. 9. Написать функцию <code>int fib(int n)</code>, вычисляющую n-е число Фибоначчи. С ее помощью вывести все числа Фибоначчи от 1 до 10. 10. Написать функцию <code>int get_prime(int i)</code>, возвращающую i-е по счету простое число. Для определения, является ли число 106 простым, использовать функцию <code>is_prime()</code>. Написать программу, которая вводит с клавиатуры число n и печатает n-ое простое число. 11. Написать функцию <code>void prime_divisors(int n)</code>, печатающую все простые делители числа n. Для определения, является ли число простым, использовать <code>is_prime()</code>. Написать программу, которая вводит с клавиатуры число n и печатает все его простые делители. 12. Написать функцию <code>int factorial_2(int n)</code>, вычисляющую двойной факториал: $-N!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot N$, если N — нечетное; $-N!! = 2 \cdot 4 \cdot 6 \cdot \dots \cdot N$, если N — четное. С помощью этой функции найти двойной факториал числа, вводимого пользователем с клавиатуры. 13. Написать функцию <code>int nod(int a, int b)</code>, вычисляющую наибольший общий делитель чисел с помощью алгоритма Евклида. Используя функцию <code>nod</code>, написать функцию <code>int nok(int a, int b)</code>, вычисляющую наименьшее общее кратное по формуле: $\text{НОК}(aa, bb) = aa \cdot bb / \text{НОД}(aa, bb)$. Наименьшим общим кратным (НОК) двух чисел a и b называется наименьшее число, которое делится и на a, и на b без остатка. Используя эти функции, написать программу, выводящую НОД и НОК введенных пользователем чисел. 14. Метод «переверни и сложи» заключается в следующем: берется целое положительное число n, все его цифры переворачиваются в обратном порядке (например, 123 становится 321) и полученное число прибавляется к n. Если получился палиндром, то процесс останавливается, иначе продлевается то же самое с полученной суммой. Процесс повторяется до тех пор, пока не получится палиндром. Ваша задача написать программу, которая спрашивает у пользователя число n и вычисляет по данному числу палиндром, используя метод «переверни и сложи». Программа должна напечатать получившийся палиндром и количество шагов метода «переверни и сложи», за которое он был получен.	
2.4	Строки и пользовательские структуры данных Выполните приведенные ниже задания. 1. Написать функцию, которая вычисляет количество вхождений переданного символа в строке. Например, для строки "a simple string" и символа 's' функция должна вернуть 2. Программа спрашивает у пользователя строчку и символ и печатает количество вхождений этого символа в строку. 2. Написать функцию, удаляющую первый символ из строки. 8.6 Написать функцию, проверяющую, что строка является палиндромом. 3. Написать функцию, вычисляющую количество маленьких и больших букв в строке. Например, в строке "Now I see bees, I won" больших букв – 3, маленьких – 10. Пробелы и запятые буквами не являются. 4. Написать функцию, вычисляющую количество цифр в строке. Например, в строке "sdf 2q43 jdsfhas3434" содержится 7 цифр.	4

	5. Написать функцию, вычисляющую длину первого слова в строке. Разделителем слов является пробел (строка может состоять из одного слова – в этом случае разделителей не будет, а также может быть вообще пустой). 6. Написать функцию, которая принимает на вход строчку и символ. Функцию должна выдать позицию самого последнего вхождения указанного символа в строке. 7. Написать две функции. Первая преобразует все символы строки к заглавным буквам, вторая преобразует все символы строки к строчным буквам. 8. Написать собственную функцию сравнения двух строк. Функция возвращает 1, если строки одинаковые, 0 – иначе. 9. Написать функцию сравнения двух строк без учета регистра. 9. Написать функцию, переворачивающую переданную строку задом наперед. Например, из строки "gateman" должно получиться "nametag". 10. Написать функцию, которая принимает на вход строчку и символ. Функция возвращает максимальное количество подряд идущих переданных символов в строке. Например, для строки "asdaajhys as asaaafg r" и символа 'a', нужно выдать 3. 11. Написать функцию, которая принимает на вход строчку и два символа. Функция заменяет все вхождения первого символа в строке на второй символ. Например, для строки "No lemon, no melon" и символов 'm' и 'd' должна получиться строка "No ledon, no delon". 12. Написать функцию, проверяющую, что переданную строку можно разбить на две одинаковые подстроки. Например, строку "abcabc" можно разбить на две одинаковые половины "abc". 13. Написать функцию, которая записывает строковое представление числа в массив. Число и массив передаются ей в качестве параметров. 14. Написать программу, которая выводит на экран символ, встречающийся в введенной пользователем строке чаще всего. 15. Написать программу, которая выводит на экран кратчайшее слово в строке. Слова отделяются пробелами. 16. Написать программу, которая выполняет выравнивание строки, введенной пользователем, по ширине. То есть пользователь вводит строку S и желаемую ширину N. Программа добавляет в S необходимое число пробелов между словами, так, чтобы длина строки стала равна N. Причем: – перед первым словом все пробелы удалить; – после последнего слова все пробелы удалить; – добавленные пробелы равномерно распределить между словами. Если длина строки S изначально превосходит N, то удалить из S все лишние пробелы. Если после этого длина получившейся строки попрежнему больше N, то оставить строку в таком состоянии. 17. Написать программу, которая выводит на экран слово "yes", если из букв введенной строки X можно составить введенную строку Y, при условии, что каждую букву строки X можно использовать один раз. Если это невозможно сделать, вывести "no".	
2.5	Динамические структуры данных Выполните приведенные ниже задания. Разработайте динамическую структуру для решения задачи. Последовательность чисел Фибоначчи задается по закономерности: $f_1 = 1, f_2 = 1, \dots, f_n = f_{n-1} + f_{n-2}$. Распечатайте n чисел Фибоначчи в следующем порядке: сначала все четные, затем все нечетные элементы. Разработайте динамическую структуру для решения задачи. Даны целые числа p и q. Получить все делители числа q, взаимно простые с числом p. Разработайте динамическую структуру для решения задачи. Дана последовательность из n целых чисел. Получите количество чисел последовательности, которые встречаются в ней ровно один раз. Распечатайте найденные числа. Разработайте динамическую структуру для решения задачи. На шахматной доске каждая клетка описывается двумя координатами: номером строки и номером столбца. Ферзь ставится на одну из клеток. Найдите количество и координаты всех клеток, которые "бьет" ферзь.	8
2.6	Работа с файлами Выполните приведенные ниже задания. 1. Во входном файле записаны два числа на одной строке, разделенные пробелом. В выходной файл вывести их сумму, произведение и разность – каждое значение на своей строке. Например, для 10 и 20, вывести: Sum = 30 Mul = 200 Sub = -10 2. Ввести с клавиатуры два числа, а затем вывести результат деления первого на	8

	второе с точностью до ближайшего целого, до десятых, сотых, тысячных и до десятитысячных. Для округления использовать соответствующее поле форматной строки. 3. Напишите программу, которая напечатает таблицу целых значений температуры в шкале Фаренгейта в диапазоне от 0 до 212 градусов, и соответствующие им значения в шкале Цельсия (с точностью до третьего знака после запятой). Для вычислений используйте формулу $\text{celcius} = 5.0 / 9.0 * (\text{fahrenheit} - 32)$. Результат должен быть напечатан в два столбца шириной 10 символов с выравниванием по правому краю. Перед значением температуры в шкале Цельсия должен выводиться знак как для отрицательных температур, так и для положительных. 4. В России дату и время принято представлять в виде: 04:12:57 14.03.2009 (14 мая 2009 года 4 часа 12 минут 57 секунд), в Америке та же дата и время представляются в виде: 04:12:57 AM 03/14/2009. AM используется для указания времени до полудня, PM – после, таким образом, количество часов никогда не превосходит 12. Во входном файле вам дана дата в российском формате, ваша задача записать в выходной файл ее американское представление. 5. Даны два файла с одинаковым количеством целых чисел. Написать программу, которая создает текстовый файл, содержащий эти числа, расположенные в два столбца шириной по 10 символов (в первом столбце содержатся числа из первого исходного файла, во втором – из второго файла). Столбцы разделить друг от друга при помощи разделителя « ». Числа выравниваются по правому краю столбца и отступают от него на один пробел. Например: 12 42 -123 4 0 12345 6. Дан файл, в котором написано несколько целых положительных чисел, каждое на своей строке. Известно, что количество цифр в каждом числе нечетно. Написать программу, которая выведет данные числа в выходной файл выровненные по центру. Например: 2 123 3 12345 456	
Всего		42

Отчет по практической / лабораторной работе должен соответствовать следующей структуре.

- 1) Словесная постановка задачи. В этом подразделе проводится полное описание задачи. Описывается суть задачи, анализ входящих в нее физических величин, область их допустимых значений, единицы их измерения, возможные ограничения, анализ условий при которых задача имеет решение (не имеет решения), анализ ожидаемых результатов.
- 2) Математическая модель. В этом подразделе вводятся математические описания физических величин и математическое описание их взаимодействий. Цель подраздела – представить решаемую задачу в математической формулировке.
- 3) Алгоритм решения задачи. В подразделе описывается разработка структуры алгоритма, обосновывается абстракция данных, задача разбивается на подзадачи. Схема алгоритма выполняется по ЕСПД (ГОСТ 19.003-80 и ГОСТ 19.002-80).
- 4) Листинг программы. Подраздел должен содержать текст программы на языке программирования C.
- 5) Выводы по лабораторной работе.

6. ФОРМЫ АТТЕСТАЦИИ, ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Оценка качества освоения программы

Оценка качества освоения программы осуществляется в форме зачета (защита проекта). По результатам аттестационных испытаний, выставляются отметки по двухбалльной системе: «зачтено», «не зачтено».

Проект: Разработка программного продукта (индивидуального или коллективного)

Критерии оценки проекта по разработке программного продукта:

Зачет	Четко обозначены сроки и стоимость для каждого этапа проекта. Распределены роли в команде (при групповом проекте) Выполнены требования к стилю кода, использованы оптимальные алгоритмы и структура базы данных, а также оптимальные запросы к базе данных. Интерфейс адаптивен, отсутствуют проблемы с отображением элементов, нарушения языковых норм в текстовых элементах. Программный продукт сохраняет работоспособность за определённый период времени. Простое освоение программы, подготовки данных и внесения изменений в программу. Программный продукт обеспечивает требуемый уровень производительности в соответствии с выделенными ресурсами, временем и другими условиями. Адаптация к новой среде функционирования
Незачет	Сроки и стоимость разработки приблизительны Требований к стилю кода не соблюдены Присутствуют проблем с отображением элементов, нарушением языковых норм в текстовых элементах. Сложности в освоении программы

7 ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

Учебно-методическое обеспечение программы

Основная литература:

1. Брайан Керниган, Деннис Ритчи. Язык программирования Си. Вильямс, 2013.
2. Г. Шилдт. Полный справочник по Си. Издание 4-е. «Вильямс» – 2010
3. Грег Перри, Дин Миллер Программирование на C для начинающих. М : Издательство Эксмо. - 2015. – 368 с.
4. Принц Питер, Кроуфорд Тони. «Язык C. Справочник. Полное описание языка». Издательство Вильямс, 2017. – 880 с.

Дополнительная литература:

5. Зыков, С. В. Программирование. Функциональный подход : учебник и практикум для вузов / С. В. Зыков. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 150 с. — (Высшее образование). — ISBN 978-5-534-16942-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/561744>
6. Зыков, С. В. Программирование : учебник и практикум для вузов / С. В. Зыков. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 285 с. — (Высшее образование). — ISBN 978-5-534-16031-4. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560815>
7. Черпаков, И. В. Основы программирования : учебник и практикум для вузов / И. В. Черпаков. — 2-е изд. — Москва : Издательство Юрайт, 2025. — 196 с. — (Высшее образование). — ISBN 978-5-534-18759-5. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/560807>

Материально-техническое обеспечение программы

Наименование специализированных аудиторий, кабинетов, лабораторий	Вид занятий	Наименование оборудования, программного обеспечения
Аудитория	Лекционные занятия	Учебная аудитория для проведения учебных занятий, оснащенная оборудованием и техническими средствами обучения. Переносной ноутбук, проектор, доска.
Компьютерный класс	Практические занятия	Компьютерная техника с возможностью подключения к сети «интернет».
Помещение для самостоятельной работы	Самостоятельная работа	Компьютерная техника с возможностью подключения к сети «интернет» и обеспечением доступа к электронной информационно-образовательной среде.

Перечень необходимого ПО

ПО	Mac	Linux
clang-format	17.0.5	17.0.5
libncurses5-dev	6.5	6.5
pcre	8.45	8.45
regex	-	-
valgrind	-	3.22.0
cppcheck	2.12.1	2.12.1
check	0.15.2	0.15.2
iTerm2	Newest	-
Docker	Newest	Newest
Git	Newest	Newest
CEF	0118.6.9	0118.6.9
Qt	6.7	6.7
GTK+	4.12.1	4.12.1
raygui	3.6	3.6
Gtest	1.14.0	1.14.0
dockle	Newest	Newest
VirtualBox	Newest	Newest
Wireshark	Newest	Newest
Apache Tomcat	9.0.89	9.0.89
JMeter	5.6.3	5.6.3

Elasticsearch	8.13	8.13
Postman	Newest	Newest
iperf	Newest	Newest
nginx	1.26.0	1.26.0
spawn-fcgi	1.6.4	1.6.4
fcgi	2.4.2	2.4.2
Python	3.12.3	3.12.3
PyCharm	Newest	Newest
.NET В том числе Asp. Net Core	8.0.4	8.0.4
Dotnet SDK	8.0.204	8.0.204
dotnet-ef	-	-
Oracle JDK / OpenJDK	22	22
Maven 3	3.9.6	3.9.6
VisualVM	Newest	Newest
JUnit	4.13.2	4.13.2
Java	17	17
IntelliJ IDEA	Newest	Newest
mockito-kotlin	5.3.1	5.3.1
kotlin.test	1.9	1.9
kotlinx.coroutines	1.8.0	1.8.0
Moxy	2.2.2	2.2.2
Ktor	2.3.10	2.3.10
Kotlin Exposed	0.5	0.5
SQLite	3.45.3	3.45.3
Android Studio	Newest	Newest
Android SDK	34	34
H2 Database Android	2.2.224	2.2.224
Retrofit 2	2.11.0	2.11.0
Dagger 2	2.51.1	2.51.1
Gradle	8.2	8.2
iOS SDK	Newest	-
CocoPods	Newest	Newest
SwiftUI	Newest	-

Node	20.13.0	20.13.0
Yarn	4.2.1	4.2.1
React Devtools	Newest	Newest
Angular Dev Tools	Newest	Newest
Vue Dev tools	Newest	Newest
Redux Dev tools	Newest	Newest
Go	1.22.3	1.22.3
DBeaver	Newest	Newest
PostgreSQL	14.2	14.2
pscre2	10.42	10.42

Педагогические условия:

Образовательный процесс по программе обеспечивается педагогическими кадрами, имеющими высшее образование по направлению подготовки Информатика и вычислительная техника и опыт деятельности в профессиональной сфере.

Особенности освоения программы инвалидами и лицами с ограниченными возможностями здоровья:

Для инвалидов и лиц с ограниченными возможностями учебный процесс осуществляется в соответствии с Положением «Об организации сопровождения инвалидов и лиц с ограниченными возможностями здоровья в федеральном государственном бюджетном образовательном учреждении высшего образования «Новгородский государственный университет имени Ярослава Мудрого» от 30.03.2021 г.

8. ДОПОЛНИТЕЛЬНЫЕ СВЕДЕНИЯ

Дополнительные сведения по программе «Введение в программирование на языке С»:

Сведения о разработке: впервые; новая редакция; с изменениями и/или дополнениями	–	впервые
Программа одобрена на заседании	–	кафедры информационных технологий и систем 26.12.2024, протокол № 4
Соотнесение программы к укрупненной группе направлений подготовки (код, наименование)	–	09.00.00 Информатика и вычислительная техника
Соотнесение программы к направлению подготовки (специальности) высшего образования (бакалавриата, специалитета, магистратуры, аспирантуры) или СПО (код, наименование)	–	09.03.01 Информатика и вычислительная техника

9. СВЕДЕНИЯ О РАЗРАБОТЧИКАХ

9.1. Разработчики программы:

Директор Центра подготовки кадров в сфере информационных технологий: кампус «Школа 21»

_____ А.С. Латышев

Зам. директора Центра подготовки кадров в сфере информационных технологий: кампус «Школа 21»

_____ Е.Е. Деева

Старший преподаватель кафедры информационных технологий и систем

_____ Л.Н. Цымбалюк

9.2. Руководитель структурного подразделения, разработавшего программу:

Директор Центра подготовки кадров в сфере информационных технологий: кампус «Школа 21»

_____ А.С. Латышев