

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский ядерный университет «МИФИ»
Обнинский институт атомной энергетики –
филиал федерального государственного автономного образовательного учреждения высшего образования
«Национальный исследовательский ядерный университет «МИФИ»
(ИАТЭ НИЯУ МИФИ)

Одобрено УМС ИАТЭ НИЯУ МИФИ,
Протокол №2-8/2021 От 30.08.2021

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

ПРОГРАММИРОВАНИЕ

название дисциплины

для студентов направления подготовки

09.03.01. Информатика и вычислительная техника

код и наименование направления подготовки

профиля

Вычислительные машины, комплексы, системы и сети

наименование профиля

Бакалавриат

Форма обучения: очная

г. Обнинск 2021 г.

Программа составлена в соответствии с требованиями образовательного стандарта высшего образования национального исследовательского ядерного университета «МИФИ» по направлению подготовки 09.03.01 – Информационные системы и технологии (уровень бакалавриата)

Авторы


_____ **А.Г.Царина**

доцент кафедры информационно-компьютерных дисциплин ИАТЭ НИЯУ МИФИ, кандидат физико-математических наук

Рецензент(ы)

Программа рассмотрена на заседании отделения интеллектуальных кибернетических систем (О)
(протокол № 5/7 от «30» июля 2021 г.)

Руководитель образовательной программы
09.03.01 Информатика и вычислительная техника

_____ С.О. Старков
 «30» июля 2021 г.

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы

В результате освоения ООП бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине:

Коды компетенций	Результаты освоения ООП Содержание компетенций	Перечень планируемых результатов обучения по дисциплине
ОПК-2	Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности;	знать методы отладки и решения задач на ЭВМ в различных режимах; уметь использовать прикладные системы программирования, разрабатывать основные программные документы; владеть методами выбора элементной базы для построения различных архитектур.
ОПК-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;	знать технологию разработки алгоритмов и программ; уметь ставить задачу и разрабатывать алгоритм ее решения.
ПК-3	Способен разрабатывать модели и компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии	

2. Место дисциплины в структуре ООП бакалавриата

Дисциплина **Программирование** реализуется в рамках базовой части естественнонаучного и математического цикла.

При изучении дисциплины происходит интеграция и углубление ранее полученных знаний по широкому спектру дисциплин. Для освоения дисциплины необходимы компетенции, сформированные в рамках изучения следующих дисциплин на школьном уровне: «Информатика», «Алгебра», «Геометрия».

Дисциплины и виды учебной деятельности, для которых освоение данной дисциплины необходимо как предшествующее: «Технология программирования», «Системное программирование», «Логическое программирование», «Объектно-ориентированное программирование», подготовка выпускной квалификационной работы.

Дисциплина изучается на первом курсе в первом и втором семестрах.

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

Вид работы	Форма обучения		
	Очная		
	Семестр		
	№1	№ 2	Всего

Контактная работа обучающихся с преподавателем	Количество часов на вид работы:		
	Аудиторные занятия (всего)	68	85
В том числе:			
лекции (лекции в интерактивной форме)	17 (-)	17 (-)	34 (-)
практические занятия (практические занятия в интерактивной форме)	17 (-)	34 (-)	51 (-)
лабораторные занятия	34	34	68
Промежуточная аттестация		36	36
В том числе:			
экзамен		36	36
Самостоятельная работа обучающихся	112	59	171
В том числе:			
проработка теоретического материала	56	19	75
подготовка к защите лабораторных работ	56	20	76
подготовка к экзамену		20	20
Всего (часы):	180	180	360
Всего (зачетные единицы):	5	5	10

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

4.1. Разделы дисциплины и трудоемкость по видам учебных занятий (в академических часах)

№ п/п	Наименование раздела /темы дисциплины	Виды учебной работы (в часах)				
		Очная форма обучения				
		Лек	Пр	Лаб	Внеауд	СРО
I семестр						
1	Введение в программирование	2				8
1.1.	Компьютерная технология решения задачи	1				4
1.2.	Структурное построение алгоритмов.	1				4
2	Представление основных управляющих структур программирования		2	4		8
2.1.	Стандартные типы данных		1	2		4
2.2.	Средства описания языков программирования		1	2		4
3	Основные конструкции языка программирования	2	2	6		15
3.1	Основные операторы. Функции ввода и вывода.	1	1	2		7
3.2	Конструкции ветвления и цикла	1	1	4		8
4	Структуры данных. Массивы	5	5	8		20
4.1.	Одномерные массивы. Указатели.	2	2	4		10
4.2.	Матрицы	3	3	4		10
5	Подпрограммы в С	4	4	8		23
5.1.	Функции, передача параметров	3	4	6		12
5.2.	Библиотеки функций	1		2		11
6	Строки	4	4	8		20
6.1.	Представление строк с С	2	2	4		10
6.2.	Библиотека string.h	2	2	4		10
II семестр						

7	Файлы	2	4	7
7.1.	Текстовые файлы	1	2	3
7.2.	Бинарные файлы	1	2	4
8	Структуры, абстрактные типы данных	2	4	5
9	Динамические объекты в С	2	4	7
10	Динамические структуры данных.	4	8	8
10.1.	Линейные списки	2	4	4
10.2.	Деревья	2	4	4
11	Рекурсивные определения и алгоритмы	2	4	5
12	Основы объектно-ориентированного программирования	2	6	6
13	Среда визуального проектирования для написания программ на языке программирования высокого уровня	2	4	3

Прим.: Лек – лекции, Пр – практические занятия / семинары, Лаб – лабораторные занятия, Внеауд – внеаудиторная работа, СРО – самостоятельная работа обучающихся

4.2. Содержание дисциплины, структурированное по разделам (темам)

Лекционный курс

№	Наименование раздела /темы дисциплины	Содержание
І семестр		
1	Введение в программирование	
1.1.	Компьютерная технология решения задачи	История развития ЭВМ, становление и развитие программирования. Компьютерная технология решения задачи (спецификация-метод-алгоритм-программа-тестирование-расчеты). Основные этапы решения задач на ЭВМ. Диалоговые программы; дружелюбность. Постановка задачи и спецификация программы.
1.2.	Структурное построение алгоритмов.	Критерии качества программы. Жизненный цикл программы. Введение в алгоритмизацию. Понятие алгоритма, свойства алгоритмов. Базисные алгоритмические элементы, их запись. Схемы алгоритмов и псевдокод. Способы записи алгоритма. Структурное построение алгоритмов. Основные принципы, теорема структуры и структурное программирование.
2	Представление основных управляющих структур программирования	
2.1.	Стандартные типы данных	Данные. Понятие типа данных. Стандартные типы данных. Основные скалярные типы. Структуры данных. Логический и физический уровень представления. Проектирование структур данных в программировании. Правила вывода для основных структур программирования.
2.2.	Средства описания языков программирования	Программа на языке высокого уровня. Анализ программ. Утверждения о программах. Корректность программ. Инвариантные утверждения. Языки программирования. Общие сведения. Синтаксис и семантика. Средства описания языков программирования. Метаязык.

		Основные понятия языков программирования (описания и предписания, константы и переменные, выражения, операторы, подпрограммы). Процедуры и функции.
3 Основные конструкции языка программирования		
3.1	Основные операторы. Функции ввода и вывода.	Алфавит и служебные слова. Имена и идентификаторы. Константы, переменные, значения. Понятие блока, локальные и глобальные объекты. Структура программы. Простейшие скалярные типы. Структура описательной части. Арифметические выражения. Стандартные функции. Отношения и логические выражения. Основные операторы. Присваивание, следование, ветвление (условный), цикл (с предусловием). Составной оператор. Функции ввода и вывода. Правила записи и оформления программы, роль комментариев. Структурная технология разработки программы.
3.2	Конструкции ветвления и цикла	Конструкции ветвления и цикла: оператор выбора, цикл с постусловием и цикл с параметром. Примеры. Метки и операторы перехода. Перечисляемые типы данных.
4 Структуры данных. Массивы		
4.1.	Одномерные массивы. Указатели.	Утверждения о массивах. Основные алгоритмы обработки одномерных массивов.
4.2.	Матрицы	Правила создания и использования матриц. Основные алгоритмы обработки матриц.
5 Подпрограммы в С		
5.1.	Функции, передача параметров	Процедуры и функции: описание, формальные и фактические параметры, механизм передачи параметров, примеры. Рекурсивные функции.
5.2.	Библиотеки функций	Основные стандартные функции и процедуры языка С, библиотеки функций, файлы заголовков
6 Строки		
6.1.	Представление строк с С	Способы представления строк. Основные алгоритмы обработки отдельных строк и текстов.
6.2.	Библиотека string.h	Работа со строкой через указатели. Основные алгоритмы обработки отдельных строк и текстов с использованием библиотеки string.h.
II семестр		
7 Файлы		
7.1.	Текстовые файлы	Работа с файлами. Прямой и последовательный доступ. Модель файла последовательного доступа, основные операции. Внутренние и внешние файлы. Файлы прямого доступа. Примеры работы с текстовыми файлами. Особенности организации и работы с ними.
7.2.	Бинарные файлы	Примеры работы с бинарными файлами. Особенности организации и работы с ними.
8 Структуры, абстрактные типы данных		
		Методология разработки пользовательского типа данных. Описание структура, поля структуры. Массив структур. Поиск и выборка из массива структур.
9 Динамические объекты в С		
		Указатели и ссылочный тип. Действия над ссылками и указателями.
10 Динамические структуры данных.		
10.1.	Линейные списки	Списки. Линейные списки. Основные виды и способы реализации: кольцевые списки, однонаправленные и двунаправленные списки. Основные операции:

		инициализация, вставка, удаление, поиск, формирование и просмотр. Линейный список как абстрактный тип данных. Реализация списков на С..
10.2.	Деревья	Понятие о деревьях. Основные операции: инициализация, вставка, удаление, поиск, формирование и просмотр.
11	Рекурсивные определения и алгоритмы	Способы реализации алгоритмов. Итерация и рекурсия. Примеры построения рекурсивных алгоритмов. Сравнительный анализ. Задача о Ханойской башне, оценка алгоритма. Программирование рекурсивных алгоритмов..
12	Основы объектно-ориентированного программирования	Парадигма объектно-ориентированного программирования, понятие класса как типа данных, инкапсуляция, наследование, полиморфизм, перегрузка функций и операторов, виртуальные функции, дружественные методы и классы. Создание простейших приложений с графическим интерфейсом, подобным интерфейсу операционной среды.
13	Среда визуального проектирования для написания программ на языке программирования высокого уровня	Обзор возможностей среды. Концепция проекта. Файлы заголовков, реализации, ресурсов, классов. Шаблоны приложений (одно-, много документные, модальные интерфейсы приложений). Работы с мастерами (помощниками) среды.

Лабораторные занятия

№	Наименование раздела /темы дисциплины	Содержание
I семестр		
2	Представление основных управляющих структур программирования	
2.1.	Стандартные типы данных	Исследование стандартных типов данных языка С (размер в памяти, диапазон значений, множество операций, явление переполнения)
2.2.	Средства описания языков программирования	Структура программы на языке С. Знакомство со средой программирования. Отладка и компиляция.
3	Основные конструкции языка программирования	
3.1	Основные операторы. Функции ввода и вывода.	Реализация линейных алгоритмов на программном уровне. Оформление диалоговой части программы.
3.2	Конструкции ветвления и цикла	1. Определение попадания точки в область на плоскости 2. Вычисление суммы ряда
4	Структуры данных. Массивы	
4.1.	Одномерные массивы. Указатели.	Реализация алгоритмов работы с массивами на программном уровне
4.2.	Матрицы	Реализация алгоритмов работы с массивами на программном уровне
5	Подпрограммы в С	
5.1.	Функции, передача параметров	Организация программы с оптимально оформленными подпрограммами. Передача массивов и матриц в качестве параметров.

5.2.	Библиотеки функций	Разработка собственной библиотеки функций, содержащей все необходимые функции для решения конкретной задачи на массивы или матрицы.
6 Строки		
6.1.	Представление строк с C	Написание программы, реализующий подход к посимвольной обработке строки
6.2.	Библиотека string.h	Написание программы, реализующей подход работы со строкой через указатели. Работа с текстом.
II семестр		
7 Файлы		
7.1.	Текстовые файлы	Разработка программы, реализующей чтение и модификацию текстового файла.
7.2.	Бинарные файлы	Разработка программы, реализующей создание, запись, чтение, модификацию бинарного файла.
8	Структуры, абстрактные типы данных	Разработка собственного типа данных согласно поставленной задаче. Создание библиотеки функций для работы с этим типом данных. Разработка управляющего модуля для демонстрации работы библиотеки.
9	Динамические объекты в C	Исследование формирования объектов в динамической памяти. Операции для работы с указателями и ссылками
10 Динамические структуры данных.		
10.1.	Линейные списки	Создание библиотеки для работы с одно- или двунаправленным линейным списком, а также управляющей программы согласно поставленной задаче
10.2.	Деревья	Создание библиотеки для работы с бинарными деревьями, а также управляющей программы согласно поставленной задаче.
11	Рекурсивные определения и алгоритмы	Реализация рекурсивных алгоритмов на программном уровне. Сравнение итерации и рекурсии.
12	Основы объектно-ориентированного программирования	Разработка простейшего класса. Создание объекта или группы объектов. Знакомство с графическим режимом.
13	Среда визуального проектирования для написания программ на языке программирования высокого уровня	Работа в среде Visual Studio. Создание проектов на основе диалоговых форм. Знакомство с понятием события.

Практические/семинарские занятия

№	Наименование раздела /темы дисциплины	Содержание
I семестр		
1 Введение в программирование		
1.1.	Компьютерная технология решения задачи	Решение математических и бытовых задач с выделением основных этапов решения. Определение исходных данных и результата. Построение цепочки: исходные данные – промежуточные вычисления –результат. Определение корректности задания исходной информации.
1.2.	Структурное построение алгоритмов.	Последовательные и вложенные структуры алгоритмов. Декомпозиция решаемой задачи на подзадачи

2 Представление основных управляющих структур программирования		
2.1.	Стандартные типы данных	Сопоставление блоков информации и типов данных. Исследование оптимальной системы для хранения входной информации, данных, получаемых во время выполнения алгоритма и результата.
2.2.	Средства описания языков программирования	Метаязык. Основные понятия языков программирования (описания и предписания, константы и переменные, выражения, операторы, подпрограммы).
3 Основные конструкции языка программирования		
3.1	Основные операторы. Функции ввода и вывода.	Решения задач на построение линейных алгоритмов и их реализация на языке программирования С.
3.2	Конструкции ветвления и цикла	Решения задач на построение циклических и разветвляющихся алгоритмов и их реализация на языке программирования С.
4 Структуры данных. Массивы		
4.1.	Одномерные массивы. Указатели.	Решение задач: заполнения массивов, преобразование элементов массивов, сдвиги элементов, добавление и удаление элементов.
4.2.	Матрицы	Решение задач: заполнение матрицы, работа с матрицей по строкам, по столбцам, удаление строк или столбцов, добавление строк или столбцов, проверка признаков для строк и столбцов
5 Подпрограммы в С		
5.1.	Функции, передача параметров	Реализация решения задач с помощью подпрограмм.
6 Строки		
6.1.	Представление строк с С	Решение задач на строки, используя представление строки как массива символов.
6.2.	Библиотека string.h	Решение задач с организацией работы со строкой через указатели. Основные алгоритмы обработки отдельных строк и текстов с использованием библиотеки string.h.

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

Для всех видов самостоятельной работы (проработки теоретического материала, подготовки к практическим занятиям, подготовки к контрольным испытаниям текущего контроля успеваемости, подготовки к зачету) обучающимся рекомендуется использовать:

- конспекты лекций;
- основную и дополнительную учебную литературу (см. раздел 7);
ресурсы информационно-телекоммуникационной сети «Интернет», в том числе периодические
<http://elibrary.ru/> – Научная электронная библиотека / Свободный доступ.
<http://ru.wikipedia.org/wiki/> – Википедия / Свободная энциклопедия
<http://mega.km.ru/pc/> – Энциклопедия Кирилла и Мефодия. Устройство персонального компьютера / Свободный доступ
<http://www.glossary.ru/> – Служба тематических толковых словарей / Свободный доступ
<http://www.intuit.ru/> – Интернет-университет Информационных Технологий / Свободно доступные курсы
<http://ocw.mit.edu/index.htm/> – MIT OpenCourseWare / Открытые.
в частности, при подготовке к контрольным точкам.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

6.1. Паспорт фонда оценочных средств по дисциплине

№ п/п	Контролируемые разделы (темы) дисциплины (результаты по разделам)	Код контролируемой компетенции (или её части) / и ее формулировка	Наименование оценочного средства
Текущий контроль, 1 семестр			
2	Представление основных управляющих структур программирования	ОПК-2 (знать)	Отчет по лабораторной работе №1 в электронном виде
3	Основные конструкции языка программирования	ОПК-2, ОПК-3 (знать, уметь), ПК-3	Отчет по лабораторной работе №2 в электронном виде
4	Структуры данных. Массивы	ОПК-2 (знать, уметь)	Отчет по лабораторной работе №3 в электронном виде
5	Подпрограммы в С	ОПК-2 (знать, уметь) ОПК-3 (знать)	Отчет по лабораторной работе №4 в электронном виде
6	Строки	ОПК-2 (знать, уметь) ОПК-3 (знать, уметь)	Отчет по лабораторной работе №5 в электронном виде
II семестр			
7	Файлы	ОПК-2 (знать, уметь) ОПК-3 (знать, уметь, владеть)	Отчет по лабораторной работе №6 в электронном виде
8	Структуры, абстрактные типы данных	ОПК-2 (знать, уметь) ОПК-3 (знать, уметь, владеть)	Отчет по лабораторной работе №7 в электронном виде
9	Динамические объекты в С	ОПК-2 (знать, уметь) ОПК-3 (знать, уметь, владеть)	Отчет по лабораторной работе №8 в электронном виде
10	Динамические структуры данных.	ОПК-2 (знать, уметь) ОПК-3 (знать, уметь, владеть)	Отчет по лабораторной работе №9 в электронном виде
11	Рекурсивные определения и алгоритмы	ОПК-2 (знать, уметь) ОПК-3 (знать, уметь, владеть)	Отчет по лабораторной работе №10 в электронном виде
12	Основы объектно-ориентированного программирования	ОПК-2 (знать, уметь) ОПК-3 (знать, уметь, владеть)	Отчет по лабораторной работе №11 в электронном виде
13	Среда визуального проектирования для написания программ на языке программирования высокого уровня	ОПК-2 (знать, уметь) ОПК-3 (знать, уметь, владеть) ПК-3	Отчет по лабораторной работе №12 в электронном виде
Промежуточный контроль, 1 семестр			
14.	Условные и циклические инструкции	ОПК-2 (знать, уметь)	Контрольная работа №1
15.	Массивы и матрицы. Функции.	ОПК-2 (знать, уметь)	Контрольная работа №2
Промежуточный контроль, 2 семестр			
14.	Динамические списки	ОПК-2 (знать, уметь)	Контрольная работа №3

		ОПК-3 (знать, уметь, владеть)	
15.	Рекурсия	ОПК-2 (знать, уметь) ОПК-3(знать, уметь, владеть)	Контрольная работа №4

6.2. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы

6.2.1 Экзамен или зачёт

6.2.1.1. Вопросы к экзамену

1. Структура программы на C: обязательные и вариативные компоненты.
2. Базовые конструкции алгоритмов, блок-схема и их синтаксическая реализация на языке C.
3. Понятие типа данных, классификация типов данных
4. Целые типы данных: операции, представление в памяти, диапазоны значений.
5. Вещественные типы данных: операции, представление в памяти, диапазоны значений.
6. Символьный тип данных: операции, представление в памяти, диапазоны значений.
7. Логический тип данных: операции, представление в памяти, диапазоны значений.
8. Перечисления: определение, диапазон значений, операции.
9. Тип данных указатели: область применения, действия над указателями, адресная арифметика, нулевой указатель.
10. Ссылки: определение, область применения, связь с указателями.
11. Совместимость и преобразование типов. Явное и неявное преобразование типов.
12. Организация консольного ввода и вывода на языках C/ C++. Форматный ввод/вывод.
13. Оператор выражение: определение, виды операторов.
14. Оператор присваивания. Комбинированные операторы присваивания. Операторы инкрементирования и декрементирования.
15. Математические и логические операции, операции сравнения.
16. Условная инструкция. Условный оператор.
17. Оператор множественного выбора.
18. Циклы с предусловием while, for
19. Цикл с постусловием do/while
20. Операторы перехода: break, continue, return.
21. Ввод и вывод данных. Универсальные функции ввода/вывода.
22. Одномерные массивы: представление в памяти, организация доступа к произвольному элементу массива, операция индексирования.
23. Основы работы с одномерными массивами: ввод/вывод элементов, сумма/произведение элементов, преобразование элементов.
24. Основы работы с одномерными массивами: поиск максимального и минимального элементов, поиск элемента с заданными свойствами.
25. Основы работы с одномерными массивами: добавление/удаление элементов, циклические сдвиги.
26. Основы работы с одномерными массивами: обмен значениями между двумя указанными элементами, инвертирование массива.
27. Основы работы с несколькими одномерными массивами: поэлементная сумма массивов, скалярное произведение.
28. Основы работы с несколькими одномерными массивами: объединение массивов (элементы одного массива следуют за элементами другого; элементы одного массива,

размещаются внутри другого массива; элементы массивов **a** и **b** располагаются в следующем порядке $\{a_0, b_0, a_1, b_1, a_2, b_2, \dots, a_{n-1}, b_{n-1}\}$).

29. Статическое и динамическое выделение памяти для хранения массивов на языке C/C++.
30. Основы работы с матрицей. Доступ к элементу матрицы. Простейшие алгоритмы (ввод/вывод элементов, сумма/произведение элементов в матрице, сумма элементов по строкам и столбцам)
31. Основы работы с матрицей. Сумма и произведение матриц.
32. Основы работы с матрицей. Транспонирование матрицы. Создание единичной матрицы.
33. Основы работы с матрицей. Преобразование матрицы в одномерный массив.
34. Статическое и динамическое выделение памяти для хранения матриц.
35. Функции и процедуры. Область действия переменных, Локальные, глобальные переменные. Видимость переменных.
36. Параметры функций и процедур. Передача параметров по адресу, по ссылке и по значению(C/C).
37. Разработка функций для оптимизации кода программы. Формальные и фактические параметры.
38. Передача массивов и матриц в качестве параметров функций и процедур.
39. Работа со строками. Форматы представления строк в C/C++. Функции для работы со строками и символами.
40. Библиотека для работы со строками string.h. Работа со строкой через указатели. Удаление символов пунктуации из строки. Поиск всех вхождений подстроки в строку.
41. Примеры обработки строк. Слияние строк, определение подстроки, удаление подстроки из строки, сравнение строк.
42. Представление текста в виде массива строк. Формирование текстовой переменной на основе данных из текстового файла.
43. Файлы. Определение. Создание файловых переменных. Инициация (открытие, связывание, направление ввода/вывода). Закрытие файлов.
44. Бинарные и типизированные файлы. Общие черты и отличия. Представление данных в них.
45. Текстовые файлы. Функции для работы с ними.
46. Бинарные файлы. Функции навигации по файлу.
47. Пользовательские типы данных. Структуры. Доступ к полям. Массивы структур/записей.
48. Пользовательские типы данных . Объединения. Принцип организации и работа с ними.
49. Модульная организация программы. (Модули и проекты с включением файлов заголовков в C). Правила компоновки.
50. Рекурсивные алгоритмы. Факториал. Ханойская башня.
51. Рекурсивные алгоритмы. Вычисление факториала. Числа Фибоначчи.
52. Динамические структуры данных. Двухнаправленный линейный список. Представление на логическом и физическом уровнях. Описание звена списка и самого списка на программном уровне.
53. Динамические структуры данных. Двухнаправленный линейный список. Добавление элемента в начало и конец списка.
54. Динамические структуры данных. Двухнаправленный линейный список. Удаление элемента из списка.
55. Динамические структуры данных. Двухнаправленный линейный список с целочисленными информационными полями. Создание списка и вывод его на печать
56. Динамические структуры данных. Деревья. Логическое и машинное представление. Основные операции.
57. Классы и объекты C++. Создание, инициализация данных, деструктор, методы для сохранения и извлечения значений полей объектов.

58. Классы С++. Конструкторы. Конструктор копирования.
59. Классы С++. Перегрузка унарных операций.
60. Классы С++. Перегрузка бинарных операций.
61. Классы С++. Перегрузка операции присваивания.
62. Классы С++. Реализация представления двунаправленного линейного списка с помощью классов.

На экзамене студенту предлагаются два вопроса из представленного перечня, и одна задача на знание ключевых алгоритмов курса и умение реализовать алгоритм на программном уровне.

<i>Критерии оценки</i>	<i>Оценка</i>
Студент должен – показать глубокое и прочное усвоение знаний по тематике вопросов; – построить грамотный, логический и исчерпывающий ответ; – правильно и полно формулировать определения; – привести примеры, иллюстрирующие теоретическую часть ответов; – правильно решить предложенные задачи, обосновать и объяснить результаты	36 – 40
Студент должен: – показать достаточно полное знание тематики вопросов; – дать достаточно грамотный и логический ответ; – правильно формулировать определения; – привести примеры для теоретической части ответов; – правильно решить предложенные задачи.	30 – 35
Студент должен – показать знания тематической части вопросов; – дать правильный ответ на большую часть вопросов; – знать основную рекомендованную учебную литературу по дисциплине. – решить предложенные задачи с незначительными ошибками.	25 – 29
Студент – не знает значительной части материала вопросов; – не владеет понятийным аппаратом дисциплины; – дает более 50% ошибочных ответов на вопросы; – не умеет делать выводы по излагаемому материалу, – не решены две из трёх задач.	≤ 24 (незачтено)

6.2.1.2. Вопросы к зачёту:

1. Структура программы на С: обязательные и вариативные компоненты.
2. Базовые конструкции алгоритмов, блок-схема и их синтаксическая реализация на языке С.
3. Понятие типа данных, классификация типов данных
4. Целые типы данных: операции, представление в памяти, диапазоны значений.
5. Вещественные типы данных: операции, представление в памяти, диапазоны значений.
6. Символьный тип данных: операции, представление в памяти, диапазоны значений.
7. Логический тип данных: операции, представление в памяти, диапазоны значений.
8. Перечисления: определение, диапазон значений, операции.
9. Тип данных указатели: область применения, действия над указателями, адресная арифметика, нулевой указатель.
10. Ссылки: определение, область применения, связь с указателями.
11. Совместимость и преобразование типов. Явное и неявное преобразование типов.

12. Организация консольного ввода и вывода на языках C/ C++. Форматный ввод/вывод.
13. Оператор выражение: определение, виды операторов.
14. Оператор присваивания. Комбинированные операторы присваивания. Операторы инкрементирования и декрементирования.
15. Математические и логические операции, операции сравнения.
16. Условная инструкция. Условный оператор.
17. Оператор множественного выбора.
18. Циклы с предусловием while, for
19. Цикл с постусловием do/while
20. Операторы перехода: break, continue, return.
21. Ввод и вывод данных. Универсальные функции ввода/вывода.
22. Одномерные массивы: представление в памяти, организация доступа к произвольному элементу массива, операция индексирования.
23. Основы работы с одномерными массивами: ввод/вывод элементов, сумма/произведение элементов, преобразование элементов.
24. Основы работы с одномерными массивами: поиск максимального и минимального элементов, поиск элемента с заданными свойствами.
25. Основы работы с одномерными массивами: добавление/удаление элементов, циклические сдвиги.
26. Основы работы с одномерными массивами: обмен значениями между двумя указанными элементами, инвертирование массива.
27. Основы работы с несколькими одномерными массивами: поэлементная сумма массивов, скалярное произведение.
28. Основы работы с несколькими одномерными массивами: объединение массивов (элементы одного массива следуют за элементами другого; элементы одного массива, размещаются внутри другого массива; элементы массивов **a** и **b** располагаются в следующем порядке $\{a_0, b_0, a_1, b_1, a_2, b_2, \dots, a_{n-1}, b_{n-1}\}$).
29. Статическое и динамическое выделение памяти для хранения массивов на языке C/C++.
30. Основы работы с матрицей. Доступ к элементу матрицы. Простейшие алгоритмы (ввод/вывод элементов, сумма/произведение элементов в матрице, сумма элементов по строкам и столбцам)
31. Основы работы с матрицей. Сумма и произведение матриц.
32. Основы работы с матрицей. Транспонирование матрицы. Создание единичной матрицы.
33. Основы работы с матрицей. Преобразование матрицы в одномерный массив.
34. Статическое и динамическое выделение памяти для хранения матриц.
35. Функции и процедуры. Область действия переменных, Локальные, глобальные переменные. Видимость переменных.
36. Параметры функций и процедур. Передача параметров по адресу, по ссылке и по значению(C/C).
37. Разработка функций для оптимизации кода программы. Формальные и фактические параметры.
38. Передача массивов и матриц в качестве параметров функций и процедур.
39. Работа со строками. Форматы представления строк в C/C++. Функции для работы со строками и символами.
40. Библиотека для работы со строками string.h. Работа со строкой через указатели. Удаление символов пунктуации из строки. Поиск всех вхождений подстроки в строку.
41. Примеры обработки строк. Слияние строк, определение подстроки, удаление подстроки из строки, сравнение строк.

На зачёте студенту предлагаются один вопрос из представленного перечня, одна задача на составление алгоритма и перевод его в на изучаемый язык программирования.

<i>Критерии оценки</i>	<i>Оценка</i>
Студент должен – показать глубокое и прочное усвоение знаний по тематике вопросов; – построить грамотный, логический и исчерпывающий ответ; – правильно и полно формулировать определения; – привести примеры, иллюстрирующие теоретическую часть ответов; – правильно решить предложенные задачи, обосновать и объяснить результаты	36 – 40
Студент должен: – показать достаточно полное знание тематики вопросов; – дать достаточно грамотный и логический ответ; – правильно формулировать определения; – привести примеры для теоретической части ответов; – правильно решить предложенные задачи.	30 – 35
Студент должен – показать знания тематической части вопросов; – дать правильный ответ на большую часть вопросов; – знать основную рекомендованную учебную литературу по дисциплине. – решить предложенные задачи с незначительными ошибками.	25 – 29
Студент – не знает значительной части материала вопросов; – не владеет понятийным аппаратом дисциплины; – дает более 50% ошибочных ответов на вопросы; – не умеет делать выводы по излагаемому материалу, – не решены две из трёх задач.	≤ 24 (незачтено)

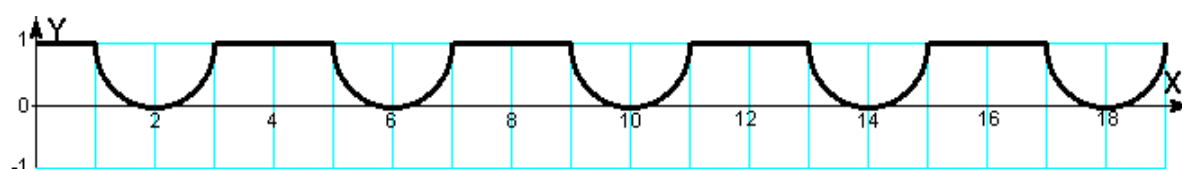
6.2.2 Промежуточный контроль

На промежуточном контроле студенту предлагается 3 или 4 задачи

а) образцы

Промежуточный контроль (Контрольная работа №1):

1. Даны действительные числа x , y . Получить $\max(x, y)$, $\min(x, y)$.
2. Дано 100 вещественных чисел, которые вводятся по одному. Найти количество положительных, отрицательных и нулевых чисел.
3. Для функции $Y = f(X)$, график которой приведен ниже, вывести на экран таблицу со значениями Y и X с шагом dX по X , где X принадлежит промежутку $[X_{\text{нач}}, X_{\text{кон}}]$.



Промежуточный контроль (Контрольная работа №2):

Написать функции, реализующие выполнение следующих ниже задач. Определить оптимальный для каждой конкретной задачи тип функции, а также список ее аргументов, передаваемых по значению и по ссылке, так чтобы с их помощью в функцию поступала вся необходимая информация, и функция возвращала все требуемые значения.

1. Дана квадратная матрица A размером $N \times N$, состоящая из действительных элементов. Сформировать массив, элементы которого равны среднему арифметическому элементов каждого из четных столбцов матрицы.
2. Дана квадратная матрица A размером $N \times N$, состоящая из действительных элементов. Поменять местами строку, содержащую максимальный элемент, со строкой, содержащей минимальный элемент.
3. Дана последовательность определенной длины, содержащая от 1 до 30 слов, в каждом из которых от 1 до 5 букв; между соседними словами - запятая, за последним словом - точка. Посчитать количество слов, которые встречаются в последовательности по одному разу

Вычислить сумму двух векторов одинаковой размерности, первый из которых вводится, а элементы второго вычисляются по формуле $b[i] = \sin(i * x)$, где $0 \leq x \leq \pi$ - вводится, $i=0,1,2,\dots,n$, n – размерность векторов. Вектора заданы в текстовом файле (каждый на своей строке, элементы введены через пробел). Результат поместить в этот же файл. Решить эту же задачу на языке Pascal, считая, что элементы векторов заданы в типизированных файлах (для каждого массива существует свой файл). Исходные данные и результат выводить в текстовый файл.

Промежуточный контроль (Контрольная работа №3):

Вариант 1	<p>Решить задачу с использованием и без использования рекурсии:</p> <p>Дан массив, состоящий из целых чисел. Создать новый массив, в который переписать только те числа, которые являются палиндромами.</p>
------------------	---

Промежуточный контроль (Контрольная работа №4):

Решить задачи, используя в качестве основной структуры данных динамический линейный однонаправленный список

1. Удалить первый элемент списка.
2. Распечатать подсписок, расположенный перед заданным элементом.

б,в) критерии оценивания компетенций (результатов) и шкала оценивания (в баллах):

Промежуточный контроль

<i>Критерии оценки</i>	<i>Оценка</i>
Студент должен – правильно решить предложенные задачи; – обосновать и объяснить результаты	29-30
Студент должен: – правильно решить предложенные задачи;	27-28
Студент должен. – решить предложенные задачи с незначительными ошибками	21 – 26
Студент должен.	20 – 18

– решить предложенные задачи с незначительными ошибками и может допустить не более двух грубых ошибок.	
Студент – не решены задачи и не может объяснить как решать.	≤ 17 (незачтено)

6.3. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Рейтинговая оценка знаний является интегральным показателем качества теоретических и практических знаний и навыков студентов по дисциплине и складывается из оценок, полученных в ходе текущего контроля и промежуточной аттестации.

Текущий контроль в семестре проводится с целью обеспечения своевременной обратной связи, для коррекции обучения, активизации самостоятельной работы студентов.

Промежуточная аттестация предназначена для объективного подтверждения и оценивания достигнутых результатов обучения после завершения изучения дисциплины.

Текущий контроль по итогам каждого лабораторного занятия.

Промежуточный контроль осуществляется два раза в семестр. В виде контрольных работ.

Результаты текущего контроля и промежуточной аттестации подводятся по шкале балльно-рейтинговой системы.

Вид контроля	Этап рейтинговой системы Оценочное средство	Балл	
		Минимум	Максимум
1 семестр			
Промежуточный контроль	Контрольная точка № 1	18	30
	Контрольная точка № 2	18	30
Итоговый	Зачёт	25	40
ИТОГО по дисциплине		60	100
2 семестр			
Промежуточный контроль	Контрольная точка № 1	18	30
	Контрольная точка № 2	18	30
Итоговый	Экзамен	25	40
ИТОГО по дисциплине		60	100

Определение бонусов и штрафов

Бонусы: поощрительные баллы студент получает к своему рейтингу в конце семестра за активную и регулярную работу на занятиях.

По Положению бонус (премиальные баллы) не может превышать **5 баллов**.

Штрафы: за несвоевременную сдачу лабораторных работ по неуважительной причине максимальная оценка за работу в семестре может быть снижена на количество баллов, равное количеству работ, сданных с опозданием.

Процедура оценивания знаний, умений, владений по дисциплине включает учет успешности по всем видам заявленных оценочных средств.

Контрольные работы проводятся на практических занятиях после четырёх лабораторных работ. Продолжительность выполнения – 45 мин.

По окончании освоения дисциплины проводится итоговая аттестация в виде зачета, что позволяет оценить совокупность приобретенных в процессе обучения компетенций. При выставлении итоговой оценки применяется балльно-рейтинговая система оценки результатов

обучения. К сдаче зачета допускаются учащиеся, которые успешно преодолели оценочные мероприятия текущего контроля.

Зачет предназначен для оценки работы обучающегося в течение всего срока изучения дисциплины и призван выявить уровень, прочность и систематичность полученных обучающимся теоретических знаний и умений приводить примеры практического использования знаний, приобретения навыков самостоятельной работы, развития творческого мышления.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

а) основная учебная литература:

- 1) Павловская Т. А. С/С++. Процедурное и объектно-ориентированное программирование : учеб. для студ. вузов / Т. А. Павловская. - СПб. : Питер, 2015. - 496 с. : ил. - (Стандарт третьего поколения) Экземпляры: ЧЗ(1), ХР(14)
- 2) Орлов С. А. Теория и практика языков программирования : учеб. для вузов - СПб. : Питер, 2014. - 688 с (24 экз.)
- 3) Delphi 7 : учеб. курс / С. Бобровский. - СПб. : Питер, 2007. - 736 с. (10 экз.)
- 4) Delphi. Программирование на языке высокого уровня : Учеб. для вузов / В. В. Фаронов. - СПб. : Питер, 2006. - 640 с. (11 экз.)
- 5) Богонин, М.Б. Языки и системы программирования. Процедурные языки программирования на примерах Pascal и Си: учебное пособие [Электронный ресурс] : учебное пособие. — Электрон. дан. — Пенза : ПензГТУ (Пензенский государственный технологический университет), 2012. — 212 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=62777 — Загл. с экрана.
- 6) Окулов, С.М. Основы программирования [Электронный ресурс] : . — Электрон. дан. — М. : "Лаборатория знаний" (ранее "БИНОМ. Лаборатория знаний"), 2012. — 340 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=8783 — Загл. с экрана.

б) дополнительная учебная литература:

- 1) Т.А. Павловская, С/С++ Программирование на языке высокого уровня: учеб. для студ. Вузов СПб. : Питер, 2011 (библиотека – 50 экз.)
- 2) Е.А. Виноградова, А.Г. Царина, Программирование на языке С/С++. Обнинск: ИАТЭ НИЯУ МИФИ, 2013. ЧЗ(5)
- 3) Виноградова Е.А., Царина А.Г. Сборник задач по программированию на языках С/С++. Обнинск, ИАТЭ, 2013 ЧЗ(5)
- 4) Виноградова Е.А., Царина А.Г. Организация работы с файлами в языках С и С++. Обнинск, ИАТЭ, 2013 ЧЗ(5)
- 5) Е.В. Крылов, В.А. Острейковский, Н. Г. Типикин Техника разработки программ. Книга 1. Программирование на языке высокого уровня. М. : Высш. Шк. 2007 (библиотека – 100 экз.)
- 6) Арипова, О.В. Программирование на языке высокого уровня: лабораторный практикум для вузов [Электронный ресурс] : / О.В. Арипова, А.Н. Гущин, О.А. Палехова. — Электрон. дан. — СПб. : БГТУ "Военмех" им. Д.Ф. Устинова (Балтийский государственный технический университет «Военмех» имени Д.Ф. Устинова), 2014. — 96 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=63671 — Загл. с экрана.
- 7) Мэйерс С. Эффективное использование С++. 55 верных способов улучшить структуру и код ваших программ [Электронный ресурс] : . — Электрон. дан. — М. : ДМК Пресс, 2008. — 300 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=1245 — Загл. с экрана.
- 8) Иванова, Г.С. Средства процедурного программирования Microsoft Visual С ++ 2008 [Электронный ресурс] : учебное пособие / Г.С. Иванова, Т.Н. Ничушкина, Р.С. Самарев. — Электрон. дан. — М. : МГТУ им. Н.Э. Баумана (Московский государственный технический университет имени Н.Э. Баумана), 2012. — 140 с. — Режим доступа: http://e.lanbook.com/books/element.php?pl1_id=52414 — Загл. с экрана.

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет» (далее – сеть «Интернет»), необходимых для освоения дисциплины

1. <http://elibrary.ru/> – Научная электронная библиотека / Свободный доступ.
2. <http://ru.wikipedia.org/wiki/> – Википедия / Свободная энциклопедия
3. <http://mega.km.ru/pc/> – Энциклопедия Кирилла и Мефодия. Устройство персонального компьютера / Свободный доступ
4. <http://www.glossary.ru/> – Служба тематических толковых словарей / Свободный доступ
5. <http://www.intuit.ru/> – Интернет-университет Информационных Технологий / Свободно доступные курсы
6. <http://ocw.mit.edu/index.htm/> – MIT OpenCourseWare / Открытые курсы Массачусетского технологического института / Свободный доступ

9. Методические указания для обучающихся по освоению дисциплины

Вид занятия	Организация деятельности студента
Лекция	<i>Вести конспект лекций</i> – кратко и последовательно фиксировать основные положения, выводы, формулировки, обобщения; пометать важные мысли, выделять ключевые слова, термины. Проверять термины и понятия с помощью поисковых систем интернета, словарей, справочников с выписыванием толкований в тетрадь. Помечать вопросы, термины, материал, который вызывает трудности, и пытаться найти ответы в рекомендуемой литературе. Если самостоятельно не удастся разобраться в проблеме – сформулировать вопрос и задать преподавателю на текущей лекции, семинарском занятии, лабораторной работе или на консультации. Для конспектов желательна отдельная тетрадь или блок страниц в общей тетради.
Лабораторная работа	<i>Выполняя лабораторную работу</i> , прежде всего, ориентироваться на содержимое тематического ЭОР и конспектировать основные положения изучаемого материала. При кодировании алгоритмов использовать ступенчатую запись, помогающую быстрее находить ошибки. Для лабораторных работ необходим электронный носитель информации (USB накопитель), на котором будут содержаться рабочие проекты и отчеты к ним.
Семинарские занятия	<i>На семинарских занятиях</i> проверяется в виде устного опроса знание теоретических вопросов разобранных на лекции, а также решаются задачи. Для подготовки к занятию студенты должны изучить материал предшествующей лекции, а также выполнить домашнее задание, которое задается на каждом семинарском занятии. Для конспектов желательна отдельная тетрадь или блок страниц в общей тетради.
Контрольное задание	<i>Контрольные задания</i> базируются на материалах лабораторных работ, семинарских занятий и лекций. Поэтому следует тщательно проработать тематический материал задания в описаниях лабораторных работ и в примерах, рассматривавшихся на лекциях, ознакомиться с основной и дополнительной литературой, включая справочные издания и интернет-источники. Программные коды заданий должны быть оформлены в соответствии с требованиями структурного программирования – ступенчатая запись, выделение управляющих структур, разумное использование пробелов для выделения структурных элементов и т.п. Работа по усмотрению преподавателя может выполняться на листах формата А4, в отдельных тетрадях для контрольных работ (хранятся на кафедре) или в электронном виде. В последнем случае количество задач, выполняемых студентом, может быть сокращено.

Подготовка к зачёту	При подготовке к зачёту ориентироваться на конспекты лекций, копии ЭОР для лабораторных работ и методические рекомендации в них, рекомендуемую литературу и т.п.
Подготовка к зачёту	При подготовке к зачёту ориентироваться на конспекты лекций, копии ЭОР для лабораторных работ и методические рекомендации в них, рекомендуемую литературу и т.п.

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

10.1. Перечень информационных технологий (при необходимости)

- Компьютерное тестирование по итогам изучения ряда разделов дисциплины.
- Использование авторских ЭОР, разработанных для проведения лабораторных работ.

10.2. Перечень программного обеспечения (при необходимости)

- Операционная система Microsoft Windows XP Professional (Service Pack 3).
- Программный комплекс Microsoft Office 2007 с набором стандартных приложений (текстовый процессор Word, табличный процессор Excel, СУБД Access и пр.).
- Браузеры MS Internet Explorer, Mozilla Firefox, Google Chrome, Opera (современные версии).
- Среда разработки Visual Studio 2005 (лицензионное ПО)

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

Для преподавания дисциплины «Информатика» существуют пять компьютерных классов кафедры ИКД с локальными сетями и доступом в интернет. Аппаратные и программные средства классов позволяют обеспечить проведение практических занятий по информатике в вычислительной среде ОС Microsoft Windows.

Три компьютерных класса в аудиториях 123-1, 123-2 и 123-3, оснащенные ПК с процессорами AMD Athlon-64 X2 Dual Core 6000+ (Socket AM2) (2 МВ, 1000 МГц) и беспроводной локальной сетью, имеют 12, 8 и 8 рабочих мест соответственно.

В компьютерном классе, имеющем 10 рабочих мест в аудитории 2-613, установлены ПК с процессорами AMD Athlon X2 Dual Core BE-2300 (Socket AM2) (2×512 КВ, 1000 МГц).

В аудитории 2-614 оборудован компьютерный класс с 23-мя ПК, имеющими процессоры Intel Celeron D-325 Socket 478 (2.53 GHz, 256 К, 533МГц).

Требования к оборудованию, содержанию и режиму работы в компьютерных классах соответствуют общегосударственным санитарно-эпидемиологическим правилам и нормативам.

12. Иные сведения и (или) материалы

12.1. Перечень образовательных технологий, используемых при осуществлении образовательного процесса по дисциплине

Все лабораторные работы по курсу «Информатика» проводятся в дисплейных классах кафедры ИКД.

12.2. Формы организации самостоятельной работы обучающихся (темы, выносимые для самостоятельного изучения; вопросы для самоконтроля; типовые задания для самопроверки)

Самостоятельная работа студентов составляет значительную часть общей трудоемкости дисциплины, является важнейшим компонентом образовательного процесса. Самостоятельная работа заключается в изучении отдельных тем курса по заданию преподавателя по

рекомендуемой им учебной литературе, в подготовке к практическим занятиям, зачету.

Список тем для самостоятельного изучения:

1. Объединения. Создание структур с вариативными полями
2. Классы. Статические элементы класса. Статические поля. Статические методы.
3. Классы. Конструкторы.
4. Программирование, управляемое событиями

12.3. Краткий терминологический словарь

Термин	Определение
Библиотека стандартных подпрограмм	Совокупность подпрограмм, составленных на одном из языков программирования и удовлетворяющих единым требованиям к структуре, организации их входов и выходов, описаниям подпрограмм.
Ввод	Считывание информации с внешнего устройства в память компьютера
Вывод	Результаты работы программы, выдаваемые компьютером пользователю, другому компьютеру или во внешнюю память.
Инструментальные программные средства	Программы, используемые в ходе разработки, корректировки или развития других программ: редакторы, отладчики, вспомогательные системные программы, графические пакеты и др. По назначению близки системам программирования.
Итерационный цикл	Вид цикла, для которого число повторений операторов тела цикла заранее неизвестно. На каждом шаге вычислений происходит последовательное приближение и проверка условия достижения искомого результата. Выход из цикла осуществляется в случае выполнения заданного условия.
Логический тип	Тип данных, представляемый значениями "истина" или "ложь" ("да" или "нет"). Иногда также называется булевским в честь английского математика XIX века Джорджа Буля.
Интерпретатор	Разновидность транслятора. Переводит и выполняет программу с языка высокого уровня в машинный код строка за строкой.
Итерационный цикл	Вид цикла, для которого число повторений операторов тела цикла заранее неизвестно. На каждом шаге вычислений происходит последовательное приближение и проверка условия достижения искомого результата. Выход из цикла осуществляется в случае выполнения заданного условия.
Компилятор	Разновидность транслятора. Читает всю программу целиком, делает её перевод и создаёт законченный вариант программы на машинном языке, который затем и выполняется.
Оператор	Фраза алгоритмического языка, определяющая некоторый законченный этап обработки данных. В состав операторов входят ключевые слова, данные, выражения и др.
Подпрограмма	Самостоятельная часть программы, которая создаётся независимо от других частей и затем вызывается по имени. Когда имя подпрограммы используется в качестве оператора программы, выполняется вся группа операторов, представляющая тело подпрограммы.
Структурное программирование	Метод разработки программ, в частности, требующий разбиения программы на небольшие независимые части (модули). Обеспечивает возможность проведения строгого доказательства правильности программ, повышает уверенность в правильности конечной программы
Язык высокого уровня	Язык программирования, более близкий к естественному языку, чем машинный код или язык ассемблера. Каждый оператор в нём соответствует нескольким командам машинного кода или языка ассемблера.