

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ
имени И.Т. ТРУБИЛИНА»

Факультет прикладной информатики
Системного анализа и обработки информации



УТВЕРЖДЕНО

Декан

Замотайлова Д.А.

Протокол от 25.04.2025 № 7

**РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)
«ПРОГРАММИРОВАНИЕ»**

Уровень высшего образования: бакалавриат

Направление подготовки: 38.03.05 Бизнес-информатика

Направленность (профиль) подготовки: Управление цифровой трансформацией бизнеса

Квалификация (степень) выпускника: бакалавр

Форма обучения: очная

Год набора (приема на обучение): 2025

Срок получения образования: 4 года

Объем:
в зачетных единицах: 8 з.е.
в академических часах: 288 ак.ч.

Разработчики:

Доцент, кафедра системного анализа и обработки информации Самойленкова В.А.

Рабочая программа дисциплины (модуля) составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 38.03.05 Бизнес-информатика, утвержденного приказом Минобрнауки от 29.07.2020 № 838, с учетом трудовых функций профессиональных стандартов: "Менеджер по информационным технологиям", утвержден приказом Минтруда России от 30.08.2021 № 588н; "Специалист по информационным системам", утвержден приказом Минтруда России от 13.07.2023 № 586н; "Системный аналитик", утвержден приказом Минтруда России от 27.04.2023 № 367н.

Согласование и утверждение

№	Подразделение или коллегиальный орган	Ответственное лицо	ФИО	Виза	Дата, протокол (при наличии)
1	Факультет прикладной информатики	Председатель методической комиссии/совет а	Крамаренко Т.А.	Согласовано	21.04.2025, № 8
2		Руководитель образовательно й программы	Вострокнутов А.Е.	Согласовано	21.04.2025, № 8

1. Цель и задачи освоения дисциплины (модуля)

Цель освоения дисциплины - «Программирование» является формирование комплекса знаний об организационных, научных и методических основах разработки алгоритмов и реализация их на одном из языков программирования высокого уровня с использованием процедурного подхода разработки программ

Задачи изучения дисциплины:

- приобретение системы знаний об алгоритмизации и программировании как одной из функций деятельности информатика, позволяющей принять соответствующие решения при разработке программ;;
- организация информационной системы для широкого круга внутренних и внешних пользователей;
- формирование и представление информации, удовлетворяющей требованиям различных пользователей программного обеспечения;
- изучение методов и способов разработки алгоритмов и программ с использованием языка высокого уровня C++, Python;
- технические и программные средства реализации информационных процессов..

2. Планируемые результаты обучения по дисциплине (модулю), соотнесенные с планируемыми результатами освоения образовательной программы

Компетенции, индикаторы и результаты обучения

ОПК-3 Способен управлять процессами создания и использования продуктов и услуг в сфере информационно-коммуникационных технологий, в том числе разрабатывать алгоритмы и программы для их практической реализации

ОПК-3.1 Правильно использует методы формализованного описания решения поставленной задачи

Знать:

ОПК-3.1/Зн1 Знает методы формализованного описания решения поставленной задачи

Уметь:

ОПК-3.1/Ум1 Умеет правильно использовать методы формализованного описания решения поставленной задачи

Владеть:

ОПК-3.1/Нв1 Владеет знаниями использования методов формализованного описания решения поставленной задачи

ОПК-3.2 Способен проводить реализацию алгоритма на языках программирования, в том числе объектно-ориентированных с использованием современных инструментальных средств разработки

Знать:

ОПК-3.2/Зн1 Знает способы реализации алгоритма на языках программирования, в том числе объектно-ориентированных с использованием современных инструментальных средств разработки

Уметь:

ОПК-3.2/Ум1 Умеет проводить реализацию алгоритма на языках программирования, в том числе объектно-ориентированных с использованием современных инструментальных средств разработки

Владеть:

ОПК-3.2/Нв1 Владеет знаниями реализации алгоритма на языках программирования, в том числе объектно-ориентированных с использованием современных инструментальных средств разработки

ОПК-3.3 Применяет на практике современные методы программирования, в том числе командной разработки, проводит сборку программного продукта и контроль качества выполненных работ

Знать:

ОПК-3.3/Зн1 Знает методы программирования, в том числе командной разработки, проведения сборки программного продукта и контроля качества выполненных работ

Уметь:

ОПК-3.3/Ум1 Умеет применять современные методы программирования, в том числе командной разработки, проводить сборку программного продукта и контроль качества выполненных работ

Владеть:

ОПК-3.3/Нв1 Применяет на практике современные методы программирования, в том числе командной разработки, проводит сборку программного продукта и контроль качества выполненных работ

ОПК-3.4 Знает основы тестирования и отладки программного продукта и способен применять их на практике

Знать:

ОПК-3.4/Зн1 Знает основы тестирования и отладки программного продукта

Уметь:

ОПК-3.4/Ум1 Умеет проводить тестирование и отладку программного продукта

Владеть:

ОПК-3.4/Нв1 Владеет знаниями тестирования и отладки программного продукта и способен применять их на практике

3. Место дисциплины в структуре ОП

Дисциплина (модуль) «Программирование» относится к обязательной части образовательной программы и изучается в семестре(ах): 1, 2.

В процессе изучения дисциплины студент готовится к решению типов задач профессиональной деятельности, предусмотренных ФГОС ВО и образовательной программой.

4. Объем дисциплины (модуля) и виды учебной работы

Период обучения	Общая трудоемкость (часы)	Общая трудоемкость (ЗЕТ)	Контактная работа (часы, всего)	Внеаудиторная контактная работа (часы)	Лабораторные занятия (часы)	Лекционные занятия (часы)	Самостоятельная работа (часы)	Промежуточная аттестация (часы)
Первый семестр	144	4	65	3	30	32	52	Экзамен (27)

Второй семестр	144	4	97	3	60	34	20	Экзамен (27)
Всего	288	8	162	6	90	66	72	54

5. Содержание дисциплины (модуля)

5.1. Разделы, темы дисциплины и виды занятий

(часы промежуточной аттестации не указываются)

Наименование раздела, темы	Всего	Внеаудиторная контактная работа	Лабораторные занятия	Лекционные занятия	Самостоятельная работа	Планируемые результаты обучения, соответствующие результатам освоения программы
Раздел 1. Основы алгоритмизации и программирования	12			4	8	ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-3.4
Тема 1.1. Алгоритмизация процессов обработки данных	6			2	4	
Тема 1.2. Введение в программирование	6			2	4	
Раздел 2. Программирование на языках высокого уровня	102		30	28	44	ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-3.4
Тема 2.1. Особенности разработки программы на C++. Лексические основы языка	10		2	4	4	
Тема 2.2. Особенности разработки программы на Python. Лексические основы языка	6		2	2	2	
Тема 2.3. Реализация ветвлений	18		6	4	8	
Тема 2.4. Реализация алгоритмов с циклической структурой	18		6	4	8	
Тема 2.5. Массивы одномерные и двумерные. Строки как массив символов	20		6	6	8	
Тема 2.6. Сортировка массивов	14		4	2	8	
Тема 2.7. Структуры. Объединения. Перечисления	16		4	6	6	
Раздел 3. Промежуточная аттестация	3	3				ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-3.4
Тема 3.1. Экзамен	3	3				
Раздел 4. Разработка приложений	114		60	34	20	ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-3.4
Тема 4.1. Программирование функций	20		12	4	4	

Тема 4.2. Указатели. Динамические переменные	24		12	8	4	
Тема 4.3. Разработка приложений с использованием файлового ввода-вывода	22		12	6	4	
Тема 4.4. Основы разработки графических приложений	24		12	8	4	
Тема 4.5. Основы разработки приложений баз данных	24		12	8	4	
Раздел 5. Промежуточная аттестация	3	3				ОПК-3.1 ОПК-3.2
Тема 5.1. Экзамен	3	3				ОПК-3.3 ОПК-3.4
Итого	234	6	90	66	72	

5.2. Содержание разделов, тем дисциплин

Раздел 1. Основы алгоритмизации и программирования (Лекционные занятия - 4ч.; Самостоятельная работа - 8ч.)

Тема 1.1. Алгоритмизация процессов обработки данных (Лекционные занятия - 2ч.; Самостоятельная работа - 4ч.)

Основные понятия. Свойства алгоритмов. Способы записи алгоритмов. Запись алгоритмов в виде блок-схем. Этапы решения прикладных задач с использованием компьютера

Тема 1.2. Введение в программирование (Лекционные занятия - 2ч.; Самостоятельная работа - 4ч.)

Основные понятия программирования. История и принципы развития языков программирования. Типы языков программирования. Виды языков высокого уровня. Системы программирования

Раздел 2. Программирование на языках высокого уровня (Лабораторные занятия - 30ч.; Лекционные занятия - 28ч.; Самостоятельная работа - 44ч.)

Тема 2.1. Особенности разработки программы на C++. Лексические основы языка (Лабораторные занятия - 2ч.; Лекционные занятия - 4ч.; Самостоятельная работа - 4ч.)

Общая характеристика языка. Технология разработки программ. Примерная структура исходного модуля программы в процедурно-ориентированном стиле. Препроцессор. Директивы препроцессора. Определение главной функции. Обзор сред программирования. Алфавит, особенности языка. Классификация типов данных. Операции, выражения и операторы. Объявление переменных. Задание констант. Операторы ввода-вывода. Область видимости и время жизни переменных. Преобразование типов. Библиотека math

Тема 2.2. Особенности разработки программы на Python. Лексические основы языка (Лабораторные занятия - 2ч.; Лекционные занятия - 2ч.; Самостоятельная работа - 2ч.)

Общая характеристика языка. Технология разработки программ. Примерная структура исходного модуля программы в процедурно-ориентированном стиле. Обзор сред программирования. Алфавит, особенности языка. Классификация типов данных. Операции, выражения и операторы. Операторы ввода-вывода. Основные библиотеки.

Тема 2.3. Реализация ветвлений (Лабораторные занятия - 6ч.; Лекционные занятия - 4ч.; Самостоятельная работа - 8ч.)

Условие в программировании. Простые и составные условия. Условный оператор if(). Оператор множественного выбора switch().

Тема 2.4. Реализация алгоритмов с циклической структурой

(Лабораторные занятия - 6ч.; Лекционные занятия - 4ч.; Самостоятельная работа - 8ч.)

Понятие циклических вычислительных процессов. Цикл for(): синтаксис, особенности. Цикл с предусловием while(): синтаксис, особенности. Цикл с постусловием do(): синтаксис, особенности. Итерационные циклы. Особенности использования и способы задания условия в итерационном цикле. Вложенные циклы. Бесконечный цикл. Операторы continue и break в циклах. Функция exit().

Тема 2.5. Массивы одномерные и двумерные. Строки как массив символов

(Лабораторные занятия - 6ч.; Лекционные занятия - 6ч.; Самостоятельная работа - 8ч.)

Массив как тип данных. Определение и инициализация одномерного массива. Доступ к элементам. Определение и инициализация двумерного массива. Доступ к элементам. Строка как массив символов типа char. Строковые переменные и константы: объявление и инициализация, вывод на экран. Одномерный массив строк. Операции со строками: копирование, сравнение, поиск, добавление строки к строке, определение длины, изменение регистра букв, кодировки букв

Тема 2.6. Сортировка массивов

(Лабораторные занятия - 4ч.; Лекционные занятия - 2ч.; Самостоятельная работа - 8ч.)

Понятие сортировки. Методы сортировки массивов. Алгоритм пузырьковой сортировки. Достоинства и недостатки метода. Реализация пузырьковой сортировки числовых массивов и строк

Тема 2.7. Структуры. Объединения. Перечисления

(Лабораторные занятия - 4ч.; Лекционные занятия - 6ч.; Самостоятельная работа - 6ч.)

Понятие структуры. Особенности структур. Определение структуры. Способы описания вложенных структур. Объявление структурных переменных. Использование операции доступа к полю структуры, в том числе вложенной. Способы инициализации полей структуры, в том числе вложенной. Объявление и инициализация массива структур. Действия со структурными переменными. Объявление типа перечисление. Объявление переменной типа перечисление. Объявление типа объединение. Объявление переменной типа объединение. Присваивание объединений друг другу

Раздел 3. Промежуточная аттестация

(Внеаудиторная контактная работа - 3ч.)

Тема 3.1. Экзамен

(Внеаудиторная контактная работа - 3ч.)

Экзамен

Раздел 4. Разработка приложений

(Лабораторные занятия - 60ч.; Лекционные занятия - 34ч.; Самостоятельная работа - 20ч.)

Тема 4.1. Программирование функций

(Лабораторные занятия - 12ч.; Лекционные занятия - 4ч.; Самостоятельная работа - 4ч.)

Понятие, назначение, особенности функций. Определение, объявление (прототипы) функций. Вызов функции пользователя. Способы описания функций в C++. Использование формальных и фактических параметров (аргументов) при обмене информацией с функцией. Рекурсивные функции

Тема 4.2. Указатели. Динамические переменные

(Лабораторные занятия - 12ч.; Лекционные занятия - 8ч.; Самостоятельная работа - 4ч.)

Указатель как тип данных. Синтаксис и семантика типа данных. Получение адреса переменной, доступ к значению переменной по адресу. Адресная арифметика. Связь указателей с массивами, в том числе строками, как массивами, функциями, структурами. Использование указателей на указатели. Свободная память. Действия с динамическими объектами: создание, уничтожение, присваивание значения. Динамические массивы: объявление, обработка

Тема 4.3. Разработка приложений с использованием файлового ввода-вывода

(Лабораторные занятия - 12ч.; Лекционные занятия - 6ч.; Самостоятельная работа - 4ч.)

Основные понятия. Текстовые и двоичные файлы. Файловый ввод/вывод с помощью потоков. Дополнительные функции файлового ввода / вывода. Режимы файлов. Произвольный доступ к файлам

Тема 4.4. Основы разработки графических приложений

(Лабораторные занятия - 12ч.; Лекционные занятия - 8ч.; Самостоятельная работа - 4ч.)

Разработка проекта приложения на основе стандартов управления проектами ISO/IEC 12207. Структура проекта. Интегрированная среда разработки программ. Компонент форма - TForm. Обработка события создания формы. Обработка событий «мыши». Обработка событий нажатия клавиш клавиатуры. Использование компонентов TLabel, Tedit, Tmemo, TmainMenu, TPopupMenu. Функции перевода данных из одного типа в другой. Функции выдачи сообщений на экран. Функции ввода данных из диалоговых окон в программу. Работа с файлами с использованием функций компонент. Работа с датами. Использование средств обработки графической информации

Тема 4.5. Основы разработки приложений баз данных

(Лабораторные занятия - 12ч.; Лекционные занятия - 8ч.; Самостоятельная работа - 4ч.)

Работа с базами данных: подключение к приложению, реализация ввода, редактирования, удаления данных в таблицах. Визуализация данных таблиц. Реализация импорта/экспорта данных в MS Word и MS Excel

Раздел 5. Промежуточная аттестация

(Внеаудиторная контактная работа - 3ч.)

Тема 5.1. Экзамен

(Внеаудиторная контактная работа - 3ч.)

Промежуточная аттестация

6. Оценочные материалы текущего контроля

Раздел 1. Основы алгоритмизации и программирования

Форма контроля/оценочное средство: Задача

Вопросы/Задания:

1. Конечный набор правил, расположенных в определенном логическом порядке, позволяющий исполнителю решать любую конкретную задачу из некоторого класса однотипных задач – это

закон

алгоритм

нормативный документ

схема

2. Алгоритм должен отвечать определенным требованиям (условиям)

однозначность
конечность
своевременность
эффективность
объемность
общность
корректность

3. Графическое представление алгоритма из отдельных геометрических фигур – блоков называется
блочной схемой
графиком
блок-схемой
диаграммой

4. Любой язык программирования включает
Алфавит языка
Цифры
Лексемы (слова)
Предложения
Выражения

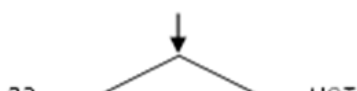
5. Укажите характеристики (атрибуты) переменной
имя переменной
псевдоним переменной
значение переменной
адрес переменной
область вызова
область видимости

6. Предложение языка программирования, которое задает полное описание действий, которые необходимо выполнить - это
Предложение языка программирования, которое задает полное описание действий, которые необходимо выполнить - это

7. Среда разработки программного обеспечения – это ...
компилятор кода
система программных средств, используемая для разработки программного обеспечения
программа, предназначенная для запуска других программ
программа, предназначенная для написания кода программ

8. Ветвление – это ...
алгоритмическая конструкция, в которой независимо от условия выполняется та или иная последовательность действий
алгоритмическая конструкция, в которой в зависимости от желания программиста выполняется строго определенная последовательность действий
алгоритмическая конструкция, в которой в зависимости от условия выполняется та или иная последовательность действий
алгоритмическая конструкция, в которой выполняется строго определенная последовательность действий, служащая для вычисления значения арифметического выражения

9. Какая алгоритмическая конструкция изображена на рисунке?
неполное ветвление
полное ветвление
цикл с предусловием
цикл с параметром
цикл с постусловием




```
`<stdio.h>`  
`<iostream>`  
`<input>`  
`<console>`
```

7. Как объявить указатель на переменную `int y`?

```
`pointer y;`  
`int* ptr = &y;`  
`ptr = y;`  
`int ptr = *y;`
```

8. Какой цикл выполняется, пока условие истинно?

```
`for`  
`while`  
`loop`  
`repeat`
```

9. Как объявить функцию, которая возвращает `int`?

```
`function int foo() {}`  
`def foo() -> int {}`  
`int foo() {}`  
`foo() int {}`
```

10. Как получить длину строки `std::string s`?

```
`s.length()`  
`len(s)`  
`sizeof(s)`  
`strlen(s)`
```

11. Какой файл создает компилятор C++ перед запуском программы?

```
` .java`  
`.exe`  
`.py`  
`.cpp`
```

12. Как вывести текст на экран в Python?

```
`echo("Hello")`  
`print("Hello")`  
`console.log("Hello")`  
`System.out.println("Hello")`
```

13. Как выйти из цикла досрочно?

```
`stop`  
`break`  
`exit`  
`return`
```

14. Как пропустить текущую итерацию цикла?

```
`skip`  
`continue`  
`next`  
`pass`
```

15. Как объявить функцию в Python?

```
`function my_func()`  
`def my_func():`  
`func my_func()`  
`void my_func()`
```

16. Как импортировать модуль `math`?

```
`import math`  
`include math`
```

```
`require math`  
`using math`
```

17. Соедините термин/код с правильным определением/результатом (Python)

- | | |
|----------------------------------|---|
| 1. <code>`print("Hello")`</code> | A. Выводит текст на экран |
| 2. <code>`x = 5`</code> | B. Присваивание переменной |
| 3. <code>`len([1, 2, 3])`</code> | C. Возвращает длину списка |
| 4. <code>`range(3)`</code> | D. Генерирует последовательность <code>`0, 1, 2`</code> |
| 5. <code>`"a" + "b"`</code> | E. Конкатенация строк (<code>`"ab"`</code>) |

18. Соедините термин/код с правильным определением/результатом (Python)

- | | |
|---------------------------------|----------------------------------|
| 1. <code>`def foo():`</code> | K. Объявление функции |
| 2. <code>`return x`</code> | L. Возврат значения из функции |
| 3. <code>`try-except`</code> | M. Обработка исключений |
| 4. <code>`*args`</code> | N. Произвольное число аргументов |
| 5. <code>`lambda x: x+1`</code> | O. Анонимная функция |

19. Какие модификаторы доступа существуют в классах?

public
private
protected
internal

20. Какие из этих исключений существуют в Python?

ValueError
IndexError
ArrayError
KeyError

Раздел 3. Промежуточная аттестация

Форма контроля/оценочное средство: Задача

Вопросы/Задания:

1. Укажите языки, которые не относятся к языкам высокого уровня

Процедурные
Функциональные
Машинно-ориентированные
Логические
Объектно-ориентированные

2. Язык программирования C++ разработал

Бьерн Страуструп
Кен Томпсон
Никлаус Вирт
Дональд Кнут

3. Укажите, какую часть исходного модуля программа на C++ должна включать обязательно

подключение заголовочных файлов библиотек (с функциями, константами и др.), определение пространства имен
объявления пользовательских символических констант, вспомогательных функций, классов и типов данных, глобальных переменных
определение главной функции
определение пользовательских вспомогательных функций и методов классов

4. Определение главной функции в C++ состоит из следующих основных частей:

заголовка определения функции
методов функции
тела определения функции

заголовка тела функции

5. Файлы с текстом программ на языке C++ имеют расширение ...

cpp
c
cs
c#

6. Выберите правильный синтаксис заголовка главной функции программы на C++

int main()
void main(<список формальных параметров>)
int main(<список формальных параметров>)
void main()

7. Пространства имен помогают

избегать конфликтов имен (функций, переменных и так далее)
сократить код
ускорить доступ к объектам (переменным, функциям и так далее)

8. Групповой проект в MS VS 20XX называется

Групповой проект в MS VS 20XX называется

9. Скомпилированные файлы проекта в MS VS 20XX хранятся в каталогах

Debug
Bin
Win32
Release

10. Алфавит языка C++ включает:

латинские буквы
арабские цифры
русские буквы
специальные символы
управляющие символы
римские цифры

11. Установите соответствие между типом данных и его записью в языке C++

(1) Логический [1] bool
(2) Символьный [2] char
(3) Вещественный [3] double
(4) Целочисленный [4] int

12. Укажите правильную последовательность команд, позволяющих ввести значение радиуса, вычислить и вывести длину окружности:

1 cout << " R= ";
2 cin >> R;
3 L=2*3.14*R;
4 cout << " L= " << L;

13. Укажите правильную последовательность команд, позволяющих ввести стороны треугольника A, B, C и вывести его площадь S

1 cin >> A >> B >> C;
2 P=(A+B+C)/2;
3 S=sqrt(P*(P-A)*(P-B)*(P-C));
4 cout << " S= " << S;

14. Укажите правильную последовательность команд, позволяющих ввести катеты прямоугольного треугольника A, B и вывести его периметр P

1 cin >> A >> B;
2 C=sqrt(A*A+B*B);
3 P=A+B+C;
4 cout << " P= " << P;

15. Выполните сопоставление знаков операций их названиям

- | | |
|------------------------------|--------------------------|
| 1 (1) Арифметические | [1] +, -, *, /, % |
| 2 (2) Отношения | [2] <, >, <=, >=, ==, != |
| 3 (3) Логические | [3] !, &&, |
| 4 (4) Присваивания | [4] = |
| 5 (5) Адресные | [5] *, & |
| 6 (6) Поразрядные логические | [6] &, , ^, <<, >>, ~ |
| 7 (7) Условная операция | [7] ? и : |
| 8 (8) Операция запятой | [8] , |

Раздел 4. Разработка приложений

Форма контроля/оценочное средство: Задача

Вопросы/Задания:

1. Чему будет равно значение переменных I и k после выполнения команд `int I=3, k=10; k=I++;`

`I = 4, k = 4`

`I = 4, k = 3`

нет правильного ответа

2. Переменная Y после выполнения команд: `X= 18; Y= X%5; Y=Y*Y;` примет значение...

Переменная Y после выполнения команд: `X= 18; Y= X%5; Y=Y*Y;` примет значение...

3. Переменные X и Y после выполнения команд: `X= 5; Y= 8; T= X; X=Y; Y= T;` примут значение...

`X=5 Y=8`

`X=8 Y=5`

`X=5 Y=5`

`X=8 Y=8`

4. Условие в программировании бывают

Одиночные

Сложные

Составные

Структурные

Простые

5. Расставьте приоритеты выполнения операций в C++ в порядке от наивысшего к низшему приоритету

1 !

2 >= <=

3 == !=

4 &&

5 ||

6. Логическое выражение, принимающее значение true, если значение X попадет в интервал [2,6]

`!(X>=2) && X<=6`

`X>=2 && X<=6`

`X<=2 || X>=6`

`2<=X<=6`

7. Сопоставьте название оператора с его синтаксисом на C++.

(1) Оператор присваивания [1] <идентификатор>=<выражение>;

(2) Условный оператор [2] `if (<выражение>) <оператор-1>; else <оператор-2>;`

(3) Составной оператор [3] `{<оператор-1>; <оператор-2>; ... <оператор-n>; }`

(4) Условная операция [4] `(<выражение>) ?<оператор-1>:<оператор-2>;`

8. Переменная X после выполнения команд: X=-2; if (X>=0) x=x*2; else x=abs(x); примет значение

Переменная X после выполнения команд: X=-2; if (X>=0) x=x*2; else x=abs(x); примет значение

9. После выполнения программы при n=1 на экран будет выведено число ...

```
int a=7;
int b=3;
int n;
cin >> n;
switch(n)
{
    case 1: cout << a+b; break;
    case 2: cout << a-b; break;
    case 3: cout << a*b; break;
    default: cout << a/b; break;
}
```

10. После выполнения программы при n=5 на экран будет выведено число...

```
int a=7;
int b=3;
int n;
cin >> n;
if (n<5)
    cout << a+b;
else
    cout << a-b;
```

11. Какой вариант оператора-переключателя является НЕверным?

1. switch(n)
{
 case 1 : cout << a+b; break;
 case 2 : cout << a-b; break;
 default: cout << a*b; break;
}
2. switch(n)
{
 case 1: case 2: cout << a+b; break;
 default: cout << a-b; break;
}
3. switch(n)
{
 case 1 : cout << a+b; break;
 case 2 : cout << a-b; break;
}
4. switch(n)
{
 case 1, 2 : cout << a+b; break;
 default: cout << a*b; break;
}

Раздел 5. Промежуточная аттестация

Форма контроля/оценочное средство: Задача

Вопросы/Задания:

1. Чем компилируемые языки отличаются от интерпретируемых?

Чем компилируемые языки отличаются от интерпретируемых?

2. Что такое переменная в программировании?

Что такое переменная в программировании?

3. Какие базовые структуры управления есть в программировании?

Какие базовые структуры управления есть в программировании?

4. Что такое синтаксис и почему он важен?

Что такое синтаксис и почему он важен?

5. Что такое IDE и зачем она нужна?

Что такое IDE и зачем она нужна?

6. Цикл do while в C++ выполняется до тех пор пока:

условие цикла остается истинным;

условие цикла остается ложным;

условие цикла остается равным нулю;

7. Установите соответствие выполняемых действий в зависимости от использования оператора в программе

(1) break [1] выполнение программы продолжается с оператора, следующего за прерванным

(2) continue [2] выполнение программы продолжается с начала прерванного оператора

8. Что будет напечатано в результате выполнения фрагмента программы?

Что будет напечатано в результате выполнения фрагмента программы?

```
for (int i = 0; i < 4; ++i)
```

```
{
    switch (i)
    {
        case 0 : cout << "0";
        case 1 : cout < "1"; continue;
        case 2 : cout << "2"; break;
        default : cout << "D"; break;
    }
    cout << ".";
}
```

0.1.2.

01.2.D.

011.2.D

Ошибка компиляции в строке 10

0112.D.

9. Какой порядковый номер (индекс) последнего элемента массива, если размер массива 19?

19

18

порядковый номер определяется программистом

10. Определить, что находит программа:

Определить, что находит программа:

```
const int n=10;
```

```
int A[n],S,i;
```

```
...
```

```
S=0;
```

```
for (i=1;i< n;i++) S+= A[i];
```

максимальное значение элементов массива

среднее арифметическое значение элементов массива

минимальное значение элементов массива

среднее квадратичное значение элементов массива

сумму элементов массива

11. Какое значение вычисляет фрагмент программы:

Какое значение вычисляет фрагмент программы:

```
float A[10][10], S=0.0;
for(int i=0;i<10;i++)
    for (int j=0;j<10;j++)
        if (i==j) S=S+A[i][j];
```

сумму элементов первого столбца матрицы

сумму элементов первой строки матрицы

сумму элементов матрицы

сумму элементов побочной диагонали матрицы

сумму элементов главной диагонали матрицы

12. Как проверить, является ли объект экземпляром класса?

```
`typeof(obj)`
`isinstance(obj, Class)`
`obj.type()`
`Class.check(obj)`
```

13. Как объявить класс в Python?

```
`class MyClass {}`
`class MyClass:`
`def class MyClass:`
`object MyClass:`
```

7. Оценочные материалы промежуточной аттестации

Первый семестр, Экзамен

Контролируемые ИДК: ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-3.4

Вопросы/Задания:

1. Что такое программирование и зачем оно нужно
2. Какие основные типы языков программирования существуют?
3. Назовите основные особенности Python
4. Опишите типы данных в Python
5. Опишите динамическую типизацию в Python
6. Опишите условный оператор `if-elif-else` в Python. Приведите пример.
7. Опишите отличие `while` от `for` в Python
8. Определение функции в Python. Опишите различие между аргументами и возвращаемым значением
9. Опишите работу со строками в Python. Приведите пример
10. Опишите понятие словаря (dictionary) в Python. Приведите пример
11. Опишите понятие `*args` и `kwargs`? Приведите пример использования.

12. Опишите импорт модулей в Python? В чем разница между ``import math`` и ``from math import sqrt``?
13. Опишите lambda-функцию в Python. Приведите пример.
14. Опишите отличие C++ от C? Назовите основные особенности C++.
15. Опишите примитивные типы данных есть в C++?
16. Опишите работу условного оператора ``if-else`` в C++. Приведите пример.
17. Опишите работу ``for``, ``while`` от ``do-while`` в C++. Приведите пример.
18. Опишите понятие функции в C++, передачу параметров по значению, ссылке и указателю?
19. Дайте понятие массива в C++, поясните отличие статического массива от динамического. Приведите пример
20. Опишите работу со строками в C++. Приведите примеры
21. Дайте понятие «указатель (``pointer``)», ссылки (``reference``) и их использование. Приведите примеры
22. Опишите работу динамического выделения памяти (``new`` и ``delete``)
23. Что такое алгоритм, типы алгоритмов? Приведите примеры
24. Опишите понятие «переменная», «константа», «литерал». Как объявить переменную в псевдокоде или языке программирования?
25. Опишите основные типы данных в программировании. Приведите примеры
26. Опишите оператор присваивания, принцип работы.
27. Опишите вложенные условия и циклы, когда они применяются. Приведите примеры.
28. Опишите операторы ``break``, ``continue``.
29. Дайте понятие функция, преимущества использование функций.
30. Что такое рекурсия? Приведите пример рекурсивной функции.
31. Дайте понятие «локальные» и «глобальные» переменные. Приведите примеры.
32. Дайте понятие «структура (``struct``)», применение структур. Приведите примеры
33. Дайте понятие список (``list``) опишите отличие от массива. Приведите примеры.

34. Жизненный цикл программного обеспечения (ПО). Назовите основные этапы.
35. Опишите разницу между водопадной (Waterfall) и гибкой (Agile) методологиями разработки?
36. Опишите принципы Agile и Scrum
37. Что такое DevOps и как он улучшает процесс разработки?
38. Опишите преимущества использования CI/CD (Continuous Integration / Continuous Deployment).
39. Опишите понятие информационной, математической, компьютерной модели.
40. Понятие операции, выражения, оператора в C++.
41. Синтаксическая структура программы на языке C++: типовые части исходного модуля.
42. Опишите статическую типизацию и преобразования типов в C++.
43. Опишите понятие «перегрузка». Приведите примеры

Второй семестр, Экзамен

Контролируемые ИДК: ОПК-3.1 ОПК-3.2 ОПК-3.3 ОПК-3.4

Вопросы/Задания:

44. Опишите определение класса и объект в Python. Приведите примеры
45. Опишите отличие метод класса (`@classmethod`) от статического метода (`@staticmethod`).
46. Опишите наследование в Python. Приведите пример.
47. Опишите основные исключения (exceptions) в Python. Приведите пример
48. Опишите работу с файлами в Python для чтения и записи. Приведите примеры
49. Что такое класс и объект в C++?
50. Какие модификаторы доступа (`public`, `private`, `protected`) существуют?
51. Опишите понятие конструктор и деструктор в C++.
52. Опишите работу наследования в C++. Приведите пример.
53. Что такое виртуальные функции и полиморфизм?

54. Опишите механизм работы исключений с помощью `'try-catch'` в C++, стандартные исключения. Приведите примеры
55. Дайте понятие класс и объект в ООП, назовите принципы ООП.
56. Опишите отличие `'public'` от `'private'` в классах. Приведите примеры
57. Дайте понятие «стек» и «куча» (`'stack'` и `'heap'`). Приведите примеры
58. Опишите основные типы ошибок в программировании.
59. Понятие элемента управления (компонента). Основные свойства и методы компонент
60. Опишите способы отладки программ
61. Для чего нужны системы контроля версий (VCS)? Назовите популярные
62. Объясните разницу между Git и GitHub/GitLab.
63. Что такое ветвление (branching) в Git? Как работает merge и rebase.
64. Опишите виды тестирования ПО.
65. Что такое отладка (debugging), инструменты отладки? .
66. Что такое шаблоны проектирования (design patterns)? Назовите несколько примеров
67. Опишите разницу между MVC, MVP и MVVM.
68. Опишите микросервисную архитектуру и чем она отличается от монолитной?
69. Что такое контейнеризация и зачем нужен Docker?
70. Какие IDE (Integrated Development Environment) вы знаете и чем они отличаются от текстовых редакторов?
71. Что такое пакетные менеджеры (npm, pip, Maven) и для чего они нужны?
72. Как работает виртуальное окружение (venv, conda, virtualenv) в Python?
73. Что такое Docker и Kubernetes? Как они используются в разработке?
74. Какие инструменты для автоматизации сборки (Make, Gradle, CMake) вы знаете?
75. Опишите назначение документирования кода, форматы документации.
76. Что такое линтинг (linting) и какие инструменты используются (ESLint, Pylint, RuboCop)?

77. Опишите принципы чистого кода (Clean Code)

78. Что такое SOLID и как эти принципы применяются в разработке

79. Основные понятия объектно-ориентированного программирования: объект, класс, поле, метод, свойства, сообщение, событие и т. д.

80. Файловые потоки. Открытие и закрытие. Режимы, типы файлов.

81. Многомерные динамические массивы. Алгоритм выделения, алгоритм освобождения памяти

82. Визуальное программирование. Понятие формы, основные события форм

8. Материально-техническое и учебно-методическое обеспечение дисциплины

8.1. Перечень основной и дополнительной учебной литературы

Основная литература

1. Агалаков,, А. А. Программирование на языке Python. Базовый уровень: учебное пособие / А. А. Агалаков,, К. И. Дементьева,. - Программирование на языке Python. Базовый уровень - Новосибирск: Сибирский государственный университет телекоммуникаций и информатики, 2024. - 117 с. - 2227-8397. - Текст: электронный // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/149539.html> (дата обращения: 08.09.2025). - Режим доступа: по подписке

2. Титов А. Н. Визуализация данных в Python. Основы работы с интерактивной библиотекой Altair: учебно-методическое пособие / Титов А. Н., Тазиева Р. Ф.. - Казань: КНИТУ, 2024. - 128 с. - 978-5-7882-3478-6. - Текст: электронный. // RuSpLAN: [сайт]. - URL: <https://e.lanbook.com/img/cover/book/477896.jpg> (дата обращения: 19.06.2025). - Режим доступа: по подписке

3. САМОЙЛЕНКОВА В. А. Алгоритмическое программирование на C++: учеб. пособие / САМОЙЛЕНКОВА В. А., Крамаренко Т. А.. - Краснодар: КубГАУ, 2025. - 137 с. - 978-5-907976-01-6. - Текст: непосредственный.

4. ИВАНОВА Е. А. Объектно-ориентированное программирование: метод. указания / ИВАНОВА Е. А., Ефанова Н. В.. - Краснодар: КубГАУ, 2022. - 34 с. - Текст: электронный. // : [сайт]. - URL: <https://edu.kubsau.ru/mod/resource/view.php?id=11422> (дата обращения: 08.09.2025). - Режим доступа: по подписке

5. Советов П. Н. Программирование на языке Питон / Советов П. Н.. - Москва: РТУ МИРЭА, 2021. - 105 с. - Текст: электронный. // RuSpLAN: [сайт]. - URL: <https://e.lanbook.com/img/cover/book/226562.jpg> (дата обращения: 19.06.2025). - Режим доступа: по подписке

Дополнительная литература

1. Лужков,, А. А. Основы работы в Python: учебно-методическое пособие / А. А. Лужков,, А. С. Тюканов,. - Основы работы в Python - Санкт-Петербург: Издательство РГПУ им. А. И. Герцена, 2022. - 76 с. - 978-5-8064-3224-8. - Текст: электронный // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/131742.html> (дата обращения: 08.09.2025). - Режим доступа: по подписке

2. Титов А. Н. Решение задач теории вероятностей и математической статистики в Python: учебно-методическое пособие / Титов А. Н., Тазиева Р. Ф.. - Казань: КНИТУ, 2022. - 144 с. - 978-5-7882-3251-5. - Текст: электронный. // RuSpLAN: [сайт]. - URL: <https://e.lanbook.com/img/cover/book/412454.jpg> (дата обращения: 19.06.2025). - Режим доступа: по подписке

3. Титов А. Н. Обработка данных в Python. Основы работы с библиотекой Pandas: учебно-методическое пособие / Титов А. Н., Тазиева Р. Ф.. - Казань: КНИТУ, 2022. - 116 с. - 978-5-7882-3164-8. - Текст: электронный. // RuSpLAN: [сайт]. - URL: <https://e.lanbook.com/img/cover/book/331013.jpg> (дата обращения: 19.06.2025). - Режим доступа: по подписке

4. Страуструп, Б. Язык программирования C++ для профессионалов: учебник / Б. Страуструп,. - Язык программирования C++ для профессионалов - Москва: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2025. - 670 с. - 978-5-4497-0922-6. - Текст: электронный // IPR SMART: [сайт]. - URL: <https://www.iprbookshop.ru/146412.html> (дата обращения: 08.09.2025). - Режим доступа: по подписке

5. Гуриков, С.Р. Основы алгоритмизации и программирования на Visual C++: Учебное пособие / С.Р. Гуриков. - 1 - Москва: ООО "Научно-издательский центр ИНФРА-М", 2022. - 515 с. - 978-5-16-107039-0. - Текст: электронный // Общество с ограниченной ответственностью «ЗНАНИУМ»: [сайт]. - URL: <https://znanium.com/cover/1039/1039154.jpg> (дата обращения: 08.09.2025). - Режим доступа: по подписке

6. Гаско, Р. Простой Python просто с нуля: Практическое пособие / Р. Гаско; . - 1 - Москва: Издательство "СОЛОН-Пресс", 2023. - 256 с. - 978-5-91359-334-4. - Текст: электронный // Общество с ограниченной ответственностью «ЗНАНИУМ»: [сайт]. - URL: <https://znanium.ru/cover/2185/2185854.jpg> (дата обращения: 08.09.2025). - Режим доступа: по подписке

8.2. Профессиональные базы данных и ресурсы «Интернет», к которым обеспечивается доступ обучающихся

Профессиональные базы данных

Не используются.

Ресурсы «Интернет»

1. <https://elib.kubsau.ru/MegaPro/web> - Мегapro
2. <https://e.lanbook.com/> - Lanbook
3. <https://znanium.com/> - Znanium.com
4. <http://www.iprbookshop.ru/> - IPRbook
5. <https://edu.kubsau.ru/> - Образовательный портал КубГАУ

8.3. Программное обеспечение и информационно-справочные системы, используемые при осуществлении образовательного процесса по дисциплине

Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине позволяют:

- обеспечить взаимодействие между участниками образовательного процесса, в том числе синхронное и (или) асинхронное взаимодействие посредством сети «Интернет»;
- фиксировать ход образовательного процесса, результатов промежуточной аттестации по дисциплине и результатов освоения образовательной программы;
- организовать процесс образования путем визуализации изучаемой информации посредством использования презентаций, учебных фильмов;
- контролировать результаты обучения на основе компьютерного тестирования.

Перечень лицензионного программного обеспечения:

- 1 Microsoft Windows - операционная система.

2 Microsoft Office (включает Word, Excel, Power Point) - пакет офисных приложений.

Перечень профессиональных баз данных и информационных справочных систем:

1 Гарант - правовая, <https://www.garant.ru/>

2 Консультант - правовая, <https://www.consultant.ru/>

3 Научная электронная библиотека eLibrary - универсальная, <https://elibrary.ru/>

Доступ к сети Интернет, доступ в электронную информационно-образовательную среду университета.

Перечень программного обеспечения

(обновление производится по мере появления новых версий программы)

Не используется.

Перечень информационно-справочных систем

(обновление выполняется еженедельно)

Не используется.

8.4. Специальные помещения, лаборатории и лабораторное оборудование

Университет располагает на праве собственности или ином законном основании материально-техническим обеспечением образовательной деятельности (помещениями и оборудованием) для реализации программы бакалавриата, специалитета, магистратуры по Блоку 1 "Дисциплины (модули)" и Блоку 3 "Государственная итоговая аттестация" в соответствии с учебным планом.

Каждый обучающийся в течение всего периода обучения обеспечен индивидуальным неограниченным доступом к электронной информационно-образовательной среде университета из любой точки, в которой имеется доступ к информационно-телекоммуникационной сети "Интернет", как на территории университета, так и вне его. Условия для функционирования электронной информационно-образовательной среды могут быть созданы с использованием ресурсов иных организаций.

Компьютерный класс

404эк

Персональный компьютер UNIVERSALD1 i5/16Gb/512GbSSD/23.8 - 1 шт.

408эк

Персональный компьютер IRU i5/16Gb/512GbSSD/23.8 - 1 шт.

9. Методические указания по освоению дисциплины (модуля)

Учебная работа по направлению подготовки осуществляется в форме контактной работы с преподавателем, самостоятельной работы обучающегося, текущей и промежуточной аттестаций, иных формах, предлагаемых университетом. Учебный материал дисциплины структурирован и его изучение производится в тематической последовательности. Содержание методических указаний должно соответствовать требованиям Федерального государственного образовательного стандарта и учебных программ по дисциплине. Самостоятельная работа студентов может быть выполнена с помощью материалов, размещенных на портале поддержки Moodle.

10. Методические рекомендации по освоению дисциплины (модуля)