МИНИСТЕРСТВО ТРАНСПОРТА РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РОССИЙСКИЙ УНИВЕРСИТЕТ ТРАНСПОРТА»
(РУТ (МИИТ)

Рабочая программа дисциплины (модуля),
как компонент образовательной программы
базового высшего образования
по специальности
10.05.01 Компьютерная безопасность,
утвержденной первым проректором ФГАОУ ВО
РУТ (МИИТ) Тимониным В.С.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Языки программирования

Специальность: 10.05.01 Компьютерная безопасность

Специализация: Информационная безопасность объектов информатизации на базе компьютерных систем

Форма обучения: Очная

Рабочая программа дисциплины (модуля) в виде электронного документа выгружена из единой корпоративной информационной системы управления университетом и соответствует оригиналу

Простая электронная подпись, выданная ФГАОУ ВО РУТ (МИИТ)
ID подписи: 2053
Подписал: заведующий кафедрой Баранов Леонид Аврамович
Дата: 01.09.2026

1. Общие сведения о дисциплине (модуле).

Целями освоения учебной дисциплины «Языки программирования» являются обучение студентов основам алгоритмизации и программирования задач на языке С и С++, приобретение практических навыков создания и отладки программ на персональных компьютерах. Основной целью изучения учебной дисциплины “Языки программирования” является формирование у студента компетенций в области программирования, необходимых при разработке системного и прикладного программного обеспечения для следующих задач профессиональной деятельности: проектная; контрольно-аналитическая; эксплуатационная. Дисциплина предназначена для получения знаний для решения следующих профессиональных задач (в соответствии с типами задач профессинальной деятельности): проектная деятельность: разработка и конфигурирование программно-аппаратных средств защиты информации; проектирование программных и аппаратных средств защиты информации в соответствии с техническим заданием с использованием средств автоматизации проектирования; контрольно-аналитическая деятельность: выполнение экспериментально-исследовательских работ при проведении сертификации программно-аппаратных средств защиты и анализ результатов; эксплуатационная деятельность: установка, наладка, тестирование и обслуживание системного и прикладного программного обеспечения; установка, наладка, тестирование и обслуживание аппаратно-программных средств обеспечения информационной безопасности компьютерных систем.

2. Планируемые результаты обучения по дисциплине (модулю).

Перечень формируемых результатов освоения образовательной программы (компетенций) в результате обучения по дисциплине (модулю):

ОПК-2 - Способен понимать устройство и историю развития транспортной системы;

ПК-6 - Способен разрабатывать модели угроз, формировать требования по защите информации в объектах информатизации на базе компьютерных систем, а также процессов их проектирования, создания и модернизации.

Обучение по дисциплине (модулю) предполагает, что по его результатам обучающийся будет:

Знать:

- программные средства системного и прикладного назначения для решения профессиональных задач

- подсистем защиты информации в операционных системах, программно-аппаратных средствах защиты информации, в прикладном и системном программном обеспечении

- последовательность действий по восстановлению работоспособности подсистем защиты информации в операционных системах, программно-аппаратных средствах защиты информации, в прикладном и системном программном обеспечении.

Уметь:

- анализировать результаты выполненных работ по восстановлению работоспособности подсистем защиты информации в операционных системах, программно-аппаратных средствах защиты информации, в прикладном и системном программном обеспечении;

- делать соответствующие выводы и строить свою деятельность в зависимости от достигнутых результатов и полученных выводов.

- Оценивать функциональные возможности аппаратных и программных средств, включая операционные системы, в составе компьютерной системы; проводит классификацию и устанавливает групповую принадлежность программного обеспечения.

- Выполнять работы по установке, настройке, администрированию и проверке работоспособности программно-аппаратные средства системного, прикладного и специального назначения в сфере профессиональной деятельности.

- выполнять управление инцидентами безопасности при функционировании программных средств системного, прикладного и специального назначения.

Владеть:

- навыками по выявлению и дифференциации нарушений работоспособности подсистем защиты информации в операционных системах, программно-аппаратных средствах защиты информации, в прикладном и системном программном обеспечении.

3. Объем дисциплины (модуля).

3.1. Общая трудоемкость дисциплины (модуля).

Общая трудоемкость дисциплины (модуля) составляет 12 з.е. (432 академических часа(ов).

3.2. Объем дисциплины (модуля) в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Тип учебных занятий	Количество часов				
	Всего	Семестр			
		№1	№2	№3	№4
Контактная работа при проведении учебных занятий (всего):	192	48	48	64	32
В том числе:					
Занятия лекционного типа	112	32	32	32	16
Занятия семинарского типа	80	16	16	32	16

3.3. Объем дисциплины (модуля) в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации составляет 240 академических часа (ов).

3.4. При обучении по индивидуальному учебному плану, в том числе при ускоренном обучении, объем дисциплины (модуля) может быть реализован полностью в форме самостоятельной работы обучающихся, а также в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении промежуточной аттестации.

4. Содержание дисциплины (модуля).

4.1. Занятия лекционного типа.

№ п/п	Тематика лекционных занятий / краткое содержание
1	Вводный урок Рассматриваемые вопросы: - Первые шаги - Переменные и арифметические выражения
2	Типы данных, операции и выражения Рассматриваемые вопросы: - Имена переменных - Типы даных и их размеры - Константы - Объявления - Арифметические операции

№ п/п	Тематика лекционных занятий / краткое содержание
	- Операции отношения и логические операции - Преобразование типов - Операции инкрементирования и декрементирования - Поразрядные операции - Операции с присваиваиванием и выражения с ними - Условные выражения - Приоритет и порядок вычисления - Указатели и алреса
3	Ввод-вывод Рассматриваемые вопросы: - Стандартные средства ввода-вывода - Форматированный вывод и функция printf - Списки аргументов пеерменной длины - Форматированный ввод и функция scanf - Обработка ошибок. Поток stderr и функция exit
4	Функции и структура программы Рассматриваемые вопросы: - Основы создания функций - Функции, возвращающие нецелые значения - Внешие переменные - Область действия - Заголовочные файлы - Статические переменные - Блочная структура - Инициализация
5	Управляющие конструкции Рассматриваемые вопросы: - Операторы и блоки - Оператор if-else - Конструкция else-if - Оператор switch. Перечисления - Циклы ? while и for. Ряды - Циклы ? do-while - Операторы break и continue - Оператор goto и метки
6	Указатели и массивы Рассматриваемые вопросы: - Указатели и адреса - Указатели и аргументы функций - Указатели и массивы - Адресная арифметика - Поиск минимального (максимального) элемента массива - Алгоритмы сортировки одномерного массива - Рекурсия - Символьные указатели и функции - Массивы указателей и указатели на указатели - Многомерные массивы - Инициализации массивов указателей - Указатели и многомерные массивы - Аргуметы командной строки - Указатели а функции

№ п/п	Тематика лекционных занятий / краткое содержание
7	Абстракция, классы и данные Рассматриваемые вопросы: - Классы - Определение и объявление классов - Unified modeling language — унифицированный язык моделирования. Разработка архитектуры приложения с использованием uml - Неявный указатель this - Область вилимости класса - Конструкторы - Деструкторы - Дружественные отношения - Статические члены класса - Управление копированием и перемещением. Правило нуля, правило трех, правило пяти - Перегрузка операторов и преобразования. Идиома copy and swap - Тесты. Test driven development (tdd) разработка через тестирование
8	Объектно-ориентированное программирование Рассматрivваемые вопросы: - Краткий обзор ооп - Определение базовых и производных классов - Преобразование и наследование - Конструкторы и функции управления копированием - Области видимости класса при наследовании - Чистые виртуальные функции - Контейнеры и наследование
9	Обработка исключительных ситуаций Рассматриваемые вопросы: - Обработка ошибок. Группировка исключений. Перехват исключений - Управление ресурсами - Исключения, не являющиеся ошибками. Спецификация исключений. Неперехваченные исключения. Исключения и эффективность. Альтернативы обработке ошибок. Стандартные исключения. Прокрутка стека
10	Шаблоны и общее программирование Рассматриваемые вопросы: - Определение шаблона - Создание экземпляра - Модели компиляции шаблона - Члены шаблона класса - Общий управлющий класс - Специализации шаблона
11	Умные указатели Рассматриваемые вопросы: - Unique_ptr - Shared_ptr - Weak_ptr
12	Итераторы Рассматриваемые вопросы: - Понятие итератора. Иерархия итераторов - Разработка собственного итератора
13	Паттерны проектирования Рассматриваемые вопросы:

№ п/п	Тематика лекционных занятий / краткое содержание
	- Порождающие паттерны - Поведенческие паттерны
14	Принципы гибкой разработки solid Рассматриваемые вопросы: - Знакомство с принципами гибкой разработки solid - Принципы kiss, yagni, dry
15	Функциональное программирование Рассматриваемые вопросы: - Введение в функциональное программирование - Первые шаги в функциональном программировании - Функциональные объекты - Средства создания новых функций из имеющихся - Чистота функций: как избежать изменемого состояния - Ленивые вычимления - Диапазоны - Функциональные структуры данных - Алгебраические типы данных и сопоставление с образцом - Монады - Метапрограммировнаие на шаблонах - Тестирование и отладка
16	Логическое программирование. Язык Пролог. Рассматриваемые вопросы: - Декларативный подход. - Факты, правила, запросы. - Механизм бэктрекинга. - Применение для задач искусственного интеллекта и формальной верификации.
17	Языки сценариев в контексте безопасности. Рассматриваемые вопросы: - Динамическая типизация: плюсы (скорость разработки) и минусы (ошибки времени выполнения). - Песочницы для скриптовых языков. - Инъекции кода при использовании eval/exec. - Встраивание скриптовых движков в приложения.
18	Тенденции развития языков программирования. Рассматриваемые вопросы: - Обзор современных требований к языкам: безопасность памяти, эргономика, производительность. - Языки будущего: влияние квантовых вычислений, верифицируемая компиляция. - Роль стандартизации.

4.2. Занятия семинарского типа.

Лабораторные работы

№ п/п	Наименование лабораторных работ / краткое содержание
1	Разработка алгоритмов и программ для разветвляющихся вычислительных процессов. В результате выполнения лабораторной работы студент получает навык разработки схем алгоритмов и программ реализации разветвляющихся вычислительных процессов.

№ п/п	Наименование лабораторных работ / краткое содержание
2	Разработка и отладка программ с разветвляющейся структурой. В результате выполнения работы студент получает навык разработки и отладка программ реализации разветвляющихся вычислительных процессов
3	Решение алгебраических и трансцендентных уравнений. В результате выполнения работы студент получает навык разработки и отладки программ решения алгебраических и трансцендентных уравнений.
4	Вычисление частичных сумм функциональных рядов. В результате выполнения лабораторной работы студент отрабатывает умение по проектированию и отладки программ вычисления частичных сумм функциональных рядов.
5	Табулирование значений функции. В результате выполнения работы студент отрабатывает умение по проектированию и отладки программ табулирования значений функции.
6	Реализация циклических вычислительных процессов. В результате выполнения лабораторной работы студент получает навык по разработки схем алгоритмов и программ реализации циклических вычислительных процессов.
7	Обработка одномерных массивов. В результате выполнения работ студент получает навык по разработки схем алгоритмов и программ обработки одномерных массивов.
8	Обработка многомерных массивов и строк. В результате выполнения лабораторной работы студент получает навык разработки схем алгоритмов и программ обработки многомерных массивов и строк.
9	Программирование с использованием функций. В результате выполнения лабораторной работы студент получает навык разработки схем алгоритмов и программ с применением функций.
10	Разработка библиотеки классов (структуры данных) с использованием шаблонов. В результате выполнения лабораторной работы студент получает навык разработки UMLдиаграмм классов, раработки библиотеки классов выбранной структуры данных с использование шаблонов, разработки тестовых классов.
11	Разработка библиотеки классов с использованием умных указателей. В результате выполнения лабораторной работы студент получает навык разработки UMLдиаграмм классов, раработки библиотеки классов выбранной структуры данных с использование умных указателей, разработки тестовых классов.
12	Разработка библиотеки классов с использованием собственных итераторов. В результате выполнения лабораторной работы студент получает навык разработки UMLдиаграмм классов, раработки библиотеки классов выбранной структуры данных с использование собственных итераторов, разработки тестовых классов.
13	Проектирование архитектуры классов (разработка UML-диаграмм). В результате выполнения лабораторной работы студент получает навык разработки UMLдиаграмм классов сложной архитектуры
14	Реализация библиотеки классов для решения систем дифференциальных уравнений (с применением паттернов, STL и функционального программирования). В результате выполнения лабораторной работы студент получает навык раработки библиотеки классов по UMLдиаграмме, разработанной в ЛР13 на примере решениясистемы дифференциальных уравнений первого порядка выбранным методом с использованием принципов гибкой раработки, изученных паттернов программирования, использования стандартной библиотеки stl. Применение функционального программирования (запрет на использование «сырых» циклов, использование диапазонов, лямбда-функций).
15	Управление памятью и предотвращение переполнения буфера. В результате выполнения работы студент умеет: выявлять потенциально опасные операции с

№ п/п	Наименование лабораторных работ / краткое содержание
	массивами и строками в коде C, применять безопасные функции (например, strncpy, snprintf) и реализовывать проверку границ массивов для предотвращения классических уязвимостей переполнения буфера.
16	Анализ и защита от уязвимостей форматной строки. В результате выполнения работы студент умеет: объяснять природу уязвимости форматной строки в C, демонстрировать возможность чтения и записи произвольных участков памяти через некорректное использование printf, а также писать защищенный код с фиксированными спецификаторами формата.
17	Принцип наименьших привилегий при работе с файлами. В результате выполнения работы студент умеет: разрабатывать на C/C++ приложения, которые работают с файлами и системными ресурсами, временно повышая привилегии только для выполнения критических операций и незамедлительно возвращая их обратно, минимизируя поверхность для атаки.
18	Разработка простого сниффера пакетов (с использованием Raw Sockets). В результате выполнения работы студент умеет: использовать низкоуровневые сокеты в Linux/Windows для захвата и парсинга сетевого трафика, идентифицировать заголовки протоколов (Ethernet, IP, TCP/UDP) и объяснять опасность несанкционированного перехвата данных.
19	Основы реверс-инжиниринга: Анализ исполняемого файла. В результате выполнения работы студент умеет: применять дизассемблеры (IDA Pro Free, Ghidra) для анализа простой бинарной программы, написанной на C/C++, находить в коде строковые константы, определять вызываемые функции и модифицировать поведение программы без исходного кода.
20	Разработка инструмента для тестирования паролей на сложность. В результате выполнения работы студент умеет: создавать на Python скрипт, оценивающий стойкость пароля (длина, использование разных регистров, цифр, спецсимволов), проверяющий его наличие в словарях распространенных паролей и выдающий рекомендации, используя регулярные выражения и работу с файлами.
21	Предотвращение SQL-инъекций. В результате выполнения работы студент умеет: разрабатывать на Java (или Python) простое веб-приложение или консольный клиент для работы с БД, демонстрировать уязвимость при конкатенации строк в SQL-запросе и исправлять её, применяя параметризованные запросы (PreparedStatement) и ORM.
22	Защита от межсайтового скриптинга (XSS). В результате выполнения работы студент умеет: создавать простое веб-приложение, уязвимое для XSS, внедрять вредоносный скрипт через пользовательский ввод, а затем реализовывать механизмы защиты: экранирование (эскейпинг) вывода и установку правильной Content Security Policy (CSP).
23	Создание сканера уязвимостей веб-форм. В результате выполнения работы студент умеет: писать на Python программу, которая обходит веб-страницу, находит все HTML-формы и автоматически отправляет в их поля тестовые пейлоады (например, типичные для XSS или SQL-инъекций) для первичной проверки защищенности.
24	Реализация простого шифрования и хеширования. В результате выполнения работы студент умеет: применять стандартные криптографические библиотеки (PyCrypto, Java Cryptography Architecture) для реализации шифрования/дешифрования симметричным алгоритмом (AES) и безопасного хеширования паролей с использованием «соли» (bcrypt, PBKDF2).
25	Разработка многопоточного сканера портов. В результате выполнения работы студент умеет: создавать эффективный сканер TCP-портов на Python (с использованием socket и threading), анализировать время отклика и интерпретировать полученные данные для построения карты сети, понимая принципы работы Nmap.

№ п/п	Наименование лабораторных работ / краткое содержание
26	Анализ и парсинг сетевого трафика (файлы .pcap). В результате выполнения работы студент умеет: использовать библиотеки (Scapy) для чтения файлов дампа трафика, фильтровать пакеты по протоколам, IP-адресам и портам, а также восстанавливать содержимое сессии (например, извлечь переданный файл или текст переписки).
27	Выявление и анализ гонок данных (Race Conditions). В результате выполнения работы студент умеет: писать многопоточную программу, в которой два потока одновременно работают с общим ресурсом (файл, переменная), вызывая состояние гонки, и применять примитивы синхронизации (мьютексы, семафоры) для устранения проблемы.
28	Анализ кода с помощью статических анализаторов. В результате выполнения работы студент умеет: настраивать и запускать инструменты статического анализа (SonarQube, PVS-Studio, Clang Static Analyzer) для заданного фрагмента кода, интерпретировать отчеты об ошибках и исправлять найденные уязвимости в соответствии с рекомендациями OWASP Top 10.
29	Автоматизация поиска файлов с конфиденциальной информацией. В результате выполнения работы студент умеет: писать скрипты для поиска в файловой системе документов, содержащих ключевые слова («пароль», «секретно», «паспорт»), и оценивать риски утечки данных, реализуя примитивный механизм DLP.
30	Моделирование атаки «Человек посередине» (MITM) в тестовой среде. В результате выполнения работы студент умеет: используя Python (или готовые утилиты типа Scapy), проводить ARP-спуфинг в локальной виртуальной сети и перехватывать незашифрованный трафик (например, HTTP-сессию) для демонстрации важности шифрования (HTTPS/TLS).

4.3. Самостоятельная работа обучающихся.

№ п/п	Вид самостоятельной работы
1	Изучение дополнительной литературы.
2	Подготовка к лабораторным работам.
3	Подготовка к промежуточной аттестации.
4	Подготовка к текущему контролю.

5. Перечень изданий, которые рекомендуется использовать при освоении дисциплины (модуля).

№ п/п	Библиографическое описание	Место доступа
1	Система контроля версий. Основы командной разработки: учебное пособие для ВУЗов. Васильева М. А., Филипченко К. М. Санкт-Петербург: Лань,- 144 с. , 2022	https://e.lanbook.com/book/261089
2	Система контроля версий. Основы командной разработки Васильева М.А., Филипченко К.М. Учебное пособие Изд. Санкт-Петербург: Лань - 144 с. - ISBN 978-5-507-44630-8 , 2022	https://reader.lanbook.com/book/261089#2

6. Перечень современных профессиональных баз данных и информационных справочных систем, которые могут использоваться при освоении дисциплины (модуля).

Официальный сайт РУТ (МИИТ) (https://www.miit.ru/).

Научно-техническая библиотека РУТ (МИИТ) (http:/library.miit.ru).

Образовательная платформа «Юрайт» (https://urait.ru/).

Общие информационные, справочные и поисковые системы «Консультант Плюс», «Гарант».

Электронно-библиотечная система издательства «Лань» (http://e.lanbook.com/).

7. Перечень лицензионного и свободно распространяемого программного обеспечения, в том числе отечественного производства, необходимого для освоения дисциплины (модуля).

Microsoft Internet Explorer (или другой браузер).

Операционная система Microsoft Windows.

Microsoft Office.

Язык программирования C++,

Среда Visual Studio .

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

Учебные аудитории для проведения учебных занятий, оснащенные компьютерной техникой и наборами демонстрационного оборудования.

9. Форма промежуточной аттестации:

Зачет в 1, 2, 3, 4 семестрах.

10. Оценочные материалы.

Оценочные материалы, применяемые при проведении промежуточной аттестации, разрабатываются в соответствии с локальным нормативным актом РУТ (МИИТ).

Авторы:

доцент, доцент, к.н. кафедры «Интеллектуальное управление и информационная безопасность в высокоавтоматизированных транспортных системах» Института железнодорожного транспорта Е.П. Балакина

Согласовано:

Заведующий кафедрой ИУиИБ Л.А. Баранов

Председатель учебно-методической комиссии Н.А. Андриянова