Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Шутов Олег Леонтьевич
Должность: Ректор
Дата подписания: 16.11.2026 14:55:24
Уникальный программный ключ:
6892313c2153d214b87fca0fd68c13fa12d41989

Разработка программно-информационных систем
09.03.04 Программная инженерия
2023 год набора

Приложение В

к основной профессиональной образовательной программе
по направлению подготовки 09.03.04 Программная инженерия,
утвержденной приказом от 07.12.2022 г. № 1-О
(в редакции приказа от 02.11.2026 г. № 87-О)

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОБРАЗОВАТЕЛЬНАЯ
ОРГАНИЗАЦИЯ ВЫСШЕГО ОБРАЗОВАНИЯ
«КУБАНСКИЙ ИНСТИТУТ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ»
(АНОО ВО «КИПО»)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.О.27 Системное программирование

Направление подготовки

09.03.04 Программная инженерия

Направленность (профиль)

Разработка программно-информационных систем

Уровень высшего образования

Бакалавриат

Квалификация

Бакалавр

Форма обучения

заочная

Год набора

2023

Рабочая программа дисциплины (модуля) Б1.О.27 «Системное программирование» предназначена для реализации основной профессиональной образовательной программы высшего образования.

Составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.04 Программная инженерия (Приказ Минобрнауки России от 19.09.2017 г. № 920 зарегистрирован в Минюсте России от 16.10.2017 г. № 48546).

СОДЕРЖАНИЕ

1 Цели и задачи изучения дисциплины(модуля)
1.1 Цель освоения дисциплины (модуля)
1.2 Задачи дисциплины (модуля)
1.3 Место дисциплины (модуля) в структуре образовательной программы
1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы
2. Структура и содержание дисциплины (модуля)
2.1 Распределение трудоёмкости дисциплины (модуля) по видам работ
2.2 Содержание дисциплины (модуля)
2.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)
3. Образовательные технологии, применяемые при освоении дисциплины (модуля)
4. Фонды оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)
4.1. Структура оценочных средств для текущего контроля и промежуточной аттестации по дисциплине (модулю)
4.2. Типовые задания для текущего контроля и вопросы (теоретические и практические) для промежуточной аттестации по дисциплине (модулю)
5. Методические указания для обучающихся по освоению дисциплины (модуля)
6. Материально-техническое и учебно-методическое обеспечение по дисциплине (модулю)

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Целью изучения дисциплины Б1.О.27 «Системное программирование» является освоение студентами системного программирования, приобретение навыков самостоятельного изучения отдельных тем дисциплины и решения типовых задач, приобретение навыков работы в современных интегрированных системах программирования для реализации программных продуктов, усвоение полученных знаний студентами, а также формирование у них мотивации к самообразованию за счет активизации самостоятельной познавательной деятельности.

1.2 Задачи дисциплины

изучение организации и принципов построения современных операционных систем и системных программ;

формирование представлений об общей методологии разработки системно-ориентированных программ с использованием современных алгоритмических языков и систем программирования;

углубленная подготовка обучающихся в области применения аппаратных и программных средств современных процессоров, предназначенных для поддержки многозадачных операционных систем.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Б1.О.27 «Системное программирование» относится к обязательной части Блока 1 «Дисциплины (модули)» учебного плана.

В соответствии с рабочим учебным планом дисциплина изучается на 3 курсе по заочным формам обучения.

Вид промежуточной аттестации: экзамен.

1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

Код и наименование компетенции	Код и наименование индикаторов достижения результатов обучения по дисциплине	Планируемые результаты обучения
ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ОПК-6.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий. ОПК-6.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ. ОПК-6.3. Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач	**Знать:** основные языки программирования и принципы работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий **Уметь:** разрабатывать алгоритмы, реализовывать их на основных языках программирования, работать с базами данных (вести базы данных и информационные хранилища), работать с современными программными средами разработки информационных систем и технологий для автоматизации

Код и наименование компетенции	Код и наименование индикаторов достижения результатов обучения по дисциплине	Планируемые результаты обучения
		бизнес-процессов, решать прикладные задачи различных классов **Владеть:** навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 5 зачетных единиц (180 час.), их распределение по видам работ представлено в таблице

Виды работ		Всего часов		
		ОФО	ОЗФО	ЗФО
Контактная работа, в том числе:		**-**	**-**	**20**
Аудиторные занятия (всего):		**-**	**-**	**20**
занятия лекционного типа		-	-	8
практические занятия		-	-	12
Иная контактная работа:				
Контрольная работа		-	-	-
Курсовая работа		-	-	-
Самостоятельная работа, в том числе:		**-**	**-**	**151**
Самоподготовка по темам (разделам) дисциплины		-	-	130
Подготовка к текущему контролю		-	-	21
Контроль:		**-**	**-**	**9**
Промежуточная аттестация (экзамен)		-	-	9
Общая трудоемкость	**час.**	**-**	**-**	**180**
	в том числе контактная работа	**-**	**-**	**20**
	зач. ед	**-**	**-**	**5**

2.2 Содержание дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы (темы) дисциплины, изучаемые на 3 курсе (заочная форма обучения)

№	Наименование темы/раздела	Количество часов				
		Всего	В том числе в виде практической подготовки	Аудиторная работа		Внеаудиторная работа (СР)
				Л	ПЗ	

1.	**Системное программирование** 1.1 Системное программирование: основные понятия и определения. 1.2 Ассемблеры	16	-	1	1	14
2.	**Регистры.** 2.1 Арифметические операции	17	-	-	1	16
3.	**Команды обработки строк.** 3.1 Обработка таблиц	21	-	1	1	19
4.	**Свойства операторов работы с экраном.** 4.1 Требования языка	17	-	1	2	14
5.	**Ввод и выполнение программ.** 5.1 Алгоритмы работы Ассемблеров	23	-	1	2	20
6.	**Логика и организация программы.** 6.1 Компоновка программ. 6.2 Выполнение программ	22	-	1	1	20
7.	**Макросредства.** 7.1 Макропроцессоры	18	-	1	1	16
8.	**Загрузчики и редакторы связей.** 8.1 Кросс-системы	17	-	1	2	14
9.	**Ошибки программирования.** 9.1 Введение в макроассемблер	20	-	1	1	18
	ИТОГО по разделам дисциплины	171	-	8	12	151
	Контрольная работа	-	-	-	-	-
	Курсовая работа	-	-	-	-	-
	Промежуточная аттестация (экзамен)	9	-	-	-	-
	Общая трудоемкость по дисциплине	180	-	8	12	151

Примечание: Л – лекции, ПЗ – практические занятия / семинары, СР – самостоятельная работа обучающегося

При изучении дисциплины могут применятся электронное обучение, дистанционные образовательные технологии в соответствии с ФГОС ВО.

2.3 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине)

Самостоятельная работа – это индивидуальная познавательная деятельность обучающегося как на аудиторных занятиях, так и во внеаудиторное время. Самостоятельная работа должна быть многогранной и иметь четко выраженную направленность на формирование конкретных компетенций.

Цель самостоятельной работы – овладение знаниями, профессиональными умениями и навыками, опытом исследовательской деятельности и обеспечение формирования профессиональных компетенций, воспитание потребности в самообразовании, ответственности и организованности, творческого подхода к решению проблем.

Самостоятельная работа обучающихся направлена на углубленное изучение разделов и тем рабочей программы. Самостоятельная работа предполагает изучение литературных источников, выполнение контрольных заданий и работ, проведение исследований разного характера. Работа основывается на анализе литературных источников и других материалов, а также реальных фактов, личных наблюдений и т.д.

Самостоятельная работа включает разнообразный комплекс видов и форм работы обучающихся:

- работа с лекционным материалом, предусматривающая проработку конспекта лекций и учебной литературы;

- поиск (подбор) и обзор литературы, электронных источников информации по заданной проблеме курса, написание реферата (доклада, эссе), исследовательской работы по заданной проблеме;

- выполнение задания по пропущенной или плохо усвоенной теме;

- изучение материала, вынесенного на самостоятельную проработку (отдельные темы, параграфы);

- подготовка к практическим занятиям;

- подготовка к промежуточной аттестации.

№ п/п	Вид учебно-методического обеспечения
1.	Методические рекомендации по самостоятельной работе обучающихся.
2.	Методические рекомендации по изучению дисциплины.
3.	Вопросы для письменного/устного собеседования, реферат, сообщение, доклад, эссе, практико-ориентированные задания, мини-кейсы, задания в виде расчетных задач, ситуационные задачи.

Задания для самостоятельной работы обучающихся по дисциплине Б1.О.27 «Системное программирование» представлены в учебно-методическом отделе.

Контроль результатов самостоятельной работы обучающихся может осуществляться в пределах времени, отведенного на обязательные учебные занятия и внеаудиторную самостоятельную работу обучающихся по дисциплине, может проходить в письменной, устной или смешанной форме.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) при изучении данной дисциплины предоставлена возможность выбора технологий обучения в зависимости от степени заболевания и осознания своей деятельности. При этом содержание программы дисциплины не изменяется, изменяются, как правило, форма обучения и образовательные технологии. Также обучающимся, имеющим инвалидность, и лицам с ограниченными возможностями здоровья созданы условия комфортного психологического климата в процессе обучения и возможности оказания помощи в установлении полноценных межличностных отношений с другими обучающимися.

3. Образовательные технологии, применяемые при освоении дисциплины (модуля)

В ходе изучения дисциплины предусмотрено использование следующих образовательных технологий: лекции, практические занятия, проблемное обучение, модульная технология, подготовка письменных аналитических работ, самостоятельная работа обучающихся.

Компетентностный подход в рамках преподавания дисциплины реализуется в использовании интерактивных технологий и активных методов (проектных методик, мозгового штурма, разбора конкретных ситуаций, анализа педагогических задач, педагогического эксперимента, иных форм) в сочетании с внеаудиторной работой.

Информационные технологии, применяемые при изучении дисциплины: использование информационных ресурсов, доступных в информационно-телекоммуникационной сети Интернет.

Адаптивные образовательные технологии, применяемые при изучении дисциплины – для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

4. Фонд оценочных средств для текущего контроля успеваемости и промежуточной аттестации по дисциплине (модулю)

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины Б1.О.27 «Системное программирование». Материалы для проведения текущего контроля успеваемости и промежуточной аттестации размещены в фонде оценочных средств по дисциплине Б1.О.27 «Системное программирование».

4.1. Структура оценочных средств для текущего контроля и промежуточной аттестации по дисциплине (модулю)

№ п/п	Код и наименование индикаторов достижения результатов обучения по дисциплине	Результаты обучения	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	ОПК-6.1. Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий.	**Знать:** основные языки программирования и принципы работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий **Уметь:** разрабатывать алгоритмы, реализовывать их на основных языках программирования, работать с базами данных (вести базы данных и информационные хранилища), работать с современными программными средами разработки информационных систем и технологий для автоматизации бизнес-процессов, решать прикладные задачи различных классов **Владеть:** навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач	Подготовка докладов/сообщений, вопросы для обсуждения по темам, задания открытого и закрытого типа	Вопросы на экзамене
2	ОПК-6.2. Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.		Подготовка докладов/сообщений, вопросы для обсуждения по темам, задания открытого и закрытого типа	Вопросы на экзамене
3	ОПК-6.3. Имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач		Подготовка докладов/сообщений, вопросы для обсуждения по темам, задания открытого и закрытого типа	Вопросы на экзамене

4.2. Типовые задания для текущего контроля и вопросы (теоретические и практические) для промежуточной аттестации по дисциплине (модулю)

Задания для текущего контроля и вопросы (теоретические и практические) для промежуточной аттестации, необходимые для оценки образовательных достижений обучающихся.

Текущий контроль успеваемости для обучающихся по заочной форме

Контрольная работа представляет собой систематическое, достаточно полное изложение авторского решения соответствующей проблемы и выполнение заданий в рамках дисциплины, которая является одним из видов текущего контроля успеваемости обучающихся заочной формы обучения.

Цели контрольной работы:

- проверка и оценка знаний обучающихся;
- закрепление практических навыков применения теоретических подходов и методов анализа на учебных примерах и задачах;
- получение информации об уровне самостоятельности и активности обучающегося, об эффективности форм и методов учебной работы.

Контрольные работы выполняются обучающимися в сроки, предусмотренные учебным планом и календарным учебным графиком.

Контрольная работа выполняется в рукописном или в печатном (компьютерном) варианте на листах формата А4 в 1 экземпляре с соблюдением установленного формата. Текст набирается шрифтом Times New Roman 12, через 1 интервал, абзацный отступ - 1,25 см, выравнивание по ширине страницы. Страница должна иметь следующие поля: левое - 25 мм, правое - 10 мм, верхнее и нижнее - 20 мм. Титульный лист содержит информацию об обучающемся выполнившем контрольную работу (ФИО обучающегося, направление подготовки, группа); наименование дисциплины; ФИО преподавателя, проверяющего работу.

Задания для контрольных работ разрабатываются преподавателем дисциплины по вариантам, которые содержат:

1) Задание в форме ответа на теоретический вопрос по теме (разделу) – объем не более 2-3 страниц;
2) Задания, составленные в форме тестов (2 задания открытого и закрытого типа, разработанные в фонде оценочных средств) или ответ на практическое задание (решить ситуацию, математическую задачу, расчетно-графическое задание и др.).

Готовая контрольная работа в электронном виде прикрепляется в электронную образовательную среду Moodle в профиль обучающегося выполнившего работу до начала сессии. Если работа в рукописном варианте, то она должна быть отсканирована и прикреплена.

Шкала и критерии оценивания контрольной работы

№ п/п	**Критерии**	**Зачтено**
Теоретический вопрос		
1	Глубина проработки материала	Основные теоретические положения по вопросу раскрыты. Имеются элементы обоснования выводов
2	Представление	Имеются элементы систематизации информации, факты применения профессиональной терминологии
3	Использование рекомендованной литературы	Основные источники рекомендованной литературы использованы
4	Грамотность изложения и качество оформления	Продемонстрирована культура речи. Соблюдены основные требования к оформлению
Выполнение тестовых заданий		

Если работа не отвечает названным критериям, выставляется оценка «не зачтено».

Разработка программно-информационных систем
09.03.04 Программная инженерия
2023 год набора

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)

Теоретические вопросы к экзамену

1. Принципы взаимодействия ОС Windows с прикладными программами. Типы данных в ОС Windows.
2. Оператор выбора (варианта, переключения) switch. Оператор разрыва break. Оператор безусловного перехода goto.
3. Основы организации памяти. Дискретность. Адресуемость. Ячейки памяти. Оперативная память. Распределение памяти в ПК. Физическая и виртуальная память. Постоянное запоминающее устройство
4. Двумерные (многомерные) массивы: определение двумерных массивов, особенности формирования двумерных массивов, особенности вывода и обработки двумерных массивов. Символьный массив
5. Общие принципы организации многозадачности в ОС. Основные понятия и определения. Планирование и диспетчеризация процессов в операционной системе
6. Условный оператор с операторными скобками. Вложенные условные операторы. Свойства объекта Форма. Создание группы «радиокнопок» (radioButton) и «чек-кнопок» (checkBox).
7. Система обработки прерываний. Классы прерываний. Приоритезация прерываний. Высокоприоритетные и низкоприоритетные обработчики прерываний
8. Операторы циклов. Оператор цикла с параметром (for). Оператор цикла с предусловием (while). Оператор цикла с постусловием (dowhile). Вспомогательные операторы, управляющие работой цикла: оператор разрыва break и оператор продолжения continue.
9. Средства обработки исключений в Visual C++. Разработка приложений с учетом безопасности исключений. Интерфейс между кодом с исключениями и без исключений. Операторы try, catch
10. Другие способы организации данных. Списки. Понятие списка. Способы определения и описания списка. Структурированные данные (структуры, struct).
11. Основы организации ввода / вывода в ПК. Системные и локальные шины. Стандарты шин. Устройства ввода/вывода. Дисковые массивы. Устройства архивации данных.
12. Работа с компонентом dataGridView. Вывод данных с помощью компонента dataGridView. Создание базы данных и привязка для последующего вывода и редактирования БД с помощью компонента dataGridView. Компонент DataSet
13. Графический и консольный интерфейсы программ. Создание элементарного графического окна.
14. Операторы разветвления алгоритма. Логические выражения. Условный оператор if. Условная операция.
15. Интерфейсы API-функций для разработки программ. API-функции для программ с выделением виртуальной памяти. API-функции для программ с проецированием файлов. API-функции для программ с выделением динамических областей
16. Приложения WindowsForms. Создание приложений при помощи визуальных компонентов. Создание формы. Компонент Form как основной контейнер для всех визуальных компонентов.
17. Принципы разработки динамических библиотек (DynamicLinkLibrary – DLL). Главная функция DllMain(). Экспортирование функций из DLL. Подключение DLL
18. Обработчики событий в WindowsForms. Присваивание и вывод результатов при помощи компонента Label. Ввод данных и вывод результатов. Условный оператор.

Условный оператор с двумя ветвями.

19. Операционная система (ОС). Виды ОС. Принципы построения ОС. Различия ОС. Состав системного программного обеспечения.

20. Основные принципы и технологии структурного программирования. Динамические и статические структуры данных. Несвязанные динамические структуры данных. Указатели. Работа с несвязанными динамическими структурами данных.

21. Принципы разработки программного кода для файлового ввода / вывода. API-функции для организации ввода / вывода. Механизмы асинхронного ввода / вывода

22. Понятие события и свойства. Основные визуальные компоненты программирования: Form, Button, Label, TextBox, ComboBox.

23. Структура и особенности реестра ОС Windows. Основные правила формирования программного кода: комментарий, формирование имен идентификаторов, форматирование текста. Понятия эффективности и оптимизации программного кода. Виды локальной оптимизации

24. Использование подпрограмм. Понятие подпрограммы. Структура процедур и функций. Обращение к подпрограмме. Механизм передачи параметров. Рекурсивные процедуры и функции.

25. Общие принципы обработки исключений в ОС Windows. Общее понятие исключительной ситуации. Виды исключительных ситуаций. Обработчики исключений

26. Применение вкладок на форме и компонентов диапазона значений. Компонент tabControl. Разработка программы с помощью вкладок. Отдельная вкладка как индивидуальный контейнер компонентов. Компонент trackBar. Компонент progressBar. Настройка и применение компонентов trackBar и progressBar.

27. Общие принципы размещения данных на дисках. Файлы. Файловая система. Система управления файлами. Организация данных на магнитных дисках и флэш-памяти. Алгоритмы поиска данных. Разделы и секторы диска. Загрузочные области

28. Запись и чтение данных из файла. Запись и чтение строк. Прямой доступ к файлу.

29. Основные принципы и технологии структурного и объектно-ориентированного программирования для ОС Windows

30. Основные этапы разработки прикладного программного обеспечения. Процесс разработки. Основные правила структурного программирования

31. Принципы разработки многопоточного приложения в ОС Windows. Основы многозадачности и многопоточности в Windows

32. Основы программирования на языке C#. Вид программы на языке C#. Простые типы данных. Оператор присваивания. Понятие арифметического выражения. Ввод данных с клавиатуры и вывод на экран

33. API-функции для реализации механизма многопоточности. Синхронизация потоков. Использование классов MFC для создания потоков

34. Особенности использования данных целого и вещественного типа. Правила согласования типов. Преобразования типов (конвертирование)

35. Структурная и неструктурная обработка исключений. Блоки с гарантированным завершением. Проверяемые исключения. Исключения, не требующие проверки

36. Регулярные структуры данных (массивы). Одномерные массивы: определение и описание массива, формирование массивов исходных данных, особенности вывода одномерных массивов.

37. Организация памяти в ОС Windows. Виртуальная память. Виртуальное адресное пространство (ВАП). Физическая память. Диспетчер виртуальной памяти (ДВП).

38. Строковые данные. Функция для работы со строками. Перенос символов между строкой и символьным массивом.

39. Технологии безопасности, реализованные в ОС Windows. Создание структуры SECURITY_ATTRIBUTES. API-функции для обеспечения безопасности Windows

40. Работа с классами. Основные способы работы с классом. Инкапсуляция переменных.

41. Обмен данными посредством буфера обмена ОС Windows. Структура и основные форматы буфера обмена. Операции с буфером обмена. Частные форматы буфера обмена

42. Формирование меню и обработка дополнительных форм. Компонент menuStrip. Использование компонента contextMenuStrip. Оформление меню. Добавление дополнительных форм Additional в проект. Обработка диалоговых окон. Окна сообщений (MessageBox). Вызов диалоговых форм.

43. Обмен данными посредством каналов и сокетов. Общие положения и классификация каналов. Виды сетевых протоколов. API-функции для работы с каналами и сокетами

44. Основы простой компьютерной графики. Рисование простых графических объектов. Изменение масштаба. Вращение в плоскости.

45. Структура и особенности реестра ОС Windows. Основные правила формирования программного кода: комментарий, формирование имен идентификаторов, форматирование текста. Понятия эффективности и оптимизации программного кода. Виды локальной оптимизации

46. Применение вкладок на форме и компонентов диапазона значений. Компонент tabControl. Разработка программы с помощью вкладок. Отдельная вкладка как индивидуальный контейнер компонентов. Компонент trackBar. Компонент progressBar. Настройка и применение компонентов trackBar и progressBar.

47. Операционная система (ОС). Виды ОС. Принципы построения ОС. Различия ОС. Состав системного программного обеспечения.

48. Основы программирования на языке C#. Вид программы на языке C#. Простые типы данных. Оператор присваивания. Понятие арифметического выражения. Ввод данных с клавиатуры и вывод на экран

49. Структурная и неструктурная обработка исключений. Блоки с гарантированным завершением. Проверяемые исключения. Исключения, не требующие проверки

50. Обработчики событий в WindowsForms. Присваивание и вывод результатов при помощи компонента Label. Ввод данных и вывод результатов. Условный оператор. Условный оператор с двумя ветвями.

Практические задания к экзамену

1. Разработать консольную программу на языке программирования C++, которая производит вывод чисел от 0 до 100000 в столбик на экран консоли, но на 70% выполнения программы, должна вызваться дополнительная функция, которая прервет выполнение основной программы, и будет выводить на экран фразу «Нажми любую кнопку» с периодичностью в 3 с. до тех пор, пока пользователь не нажмет любую кнопку. После завершения выполнения дополнительной функции, основная процедура продолжает выводить числа на экран консоли.

2. Разработать консольную программу на языке C++, которая будет производить расчет функции $f(x) = \sin(\pi * x)$. Создать обработчик исключительных ситуаций, которые могут произойти при вычислении значений функции.

3. Разработать консольное приложение на языке C++ по вводу/выводу данных в файловой системе: организовать асинхронный вывод данных в текстовый файл и открытие его в программе notepad, пока программа производит вывод на экран консоли бесконечное количество раз фразы «Hello, World».

4. Разработать консольное приложение на языке C++ по асинхронному вводу/выводу данных: организовать асинхронный вывод данных в текстовый файл и открытие его в программе notepad, пока программа производит вывод на экран консоли бесконечное количество раз фразы «Hello, World».

5. Разработать консольное приложение на языке C++, в которое импортировать функции из написанной библиотеки DLL. Библиотека DLLдолжна содержать две функции: функцию сложения двух чисел и функцию умножения трех чисел. После импорта функций в программу, их необходимо использовать в коде приложения.

6. Разработать консольное приложение на языке C++, в котором создать три потока выполнения вычислений. Вычислениями будут служить любые функции (например, производящие арифметические операции над случайными числами). Результаты вычислений выводятся бесконечно через определенный промежуток времени, приостанавливая потоки (т.е. каждый из потоков «спит» определенное время). При вводе определенной команды пользователем – программа останавливает все потоки и завершается.

7. Разработать консольное приложение на языке C++, в которое импортировать функции из написанной библиотеки DLL. Библиотека DLLдолжна содержать две функции: функцию сложения двух чисел и функцию умножения трех чисел. После импорта функций в программу, их необходимо использовать в коде приложения.

8. Разработать консольную программу на языке C++, которая с помощью API-функций будет создавать текстовый файл, записывать в него информацию и считывать данные из сохраненного файла, выводя их на экран консоли.

9. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 2 целых числа: ***a*** и ***b***. Второй модуль должен содержать функцию, которая создает двумерный массив ***m***, размерностью [***a***, ***b***] и заполняет его случайными целыми числами в интервале [-10, 10], исключая 0. Третий модуль программы должен включать в себя функцию, которая выводит на консоль исходный массив ***m*** из второго модуля и выполняет в нем поиск, а также выводит на экран консоли все найденные числа, которые больше -1.

10. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 3 целых числа: ***x***, **y**, ***z***. Второй модуль должен содержать функцию, которая создает одномерный массив ***mas***, размерностью [***x***] и заполняет его случайными целыми числами в интервале [***y***, ***z***], исключая отрицательные числа. Третий модуль программы должен включать в себя функцию, которая выводит на консоль исходный массив ***mas*** из второго модуля и выполняет в нем поиск, а также выводит на экран консоли сумму всех четных чисел, которые кратны трем и больше 5.

11. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 2 целых числа: ***x***, ***y***. Второй модуль должен содержать функцию, которая создает двумерный массив ***mas***, размерностью [***x, y***] и заполняет его случайными числами ***x*** ИЛИ ***y*** (т.е. каждый элемент массива должен содержать одно из двух чисел). Третий модуль программы должен включать в себя функцию, которая выводит на консоль исходный массив ***mas*** из второго модуля и выполняет в нем подсчет количества встречающихся чисел ***x*** и чисел ***y***. Также функция должна выполнить расчет процентного соотношения содержащихся чисел ***x*** и ***y*** в массиве и вывести на экран. За 100% взять количество элементов массива.

12. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен

запрашивать у пользователя 2 целых числа: ***a*** и ***b***. Второй модуль должен содержать функцию, которая создает двумерный массив ***m***, размерностью [***a***, ***b***] и заполняет его случайными прописными буквами латинского алфавита в интервале [A, Z]. Третий модуль программы должен включать в себя функцию, которая выводит на консоль исходный массив ***m*** из второго модуля и выполняет в нем поиск, а также выводит на экран консоли все найденные буквы, имеющие хотя бы одно повторение.

13. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 3 вещественных числа: ***x***, ***y***, ***z***. Второй модуль должен содержать функцию, которая выполняет табулирование функции $F(x) = \sin(x+1) \cdot e^{-(x^2-2)}$, где ***x*** - число, которое ввел пользователь в главном модуле; за шаг приращения взять ***y***. Количество действий табулирования равно ***z***. Процесс табулирования функции необходимо выводить на экран консоли. Третий модуль программы должен включать в себя функцию, которая выводит на консоль максимальное число из трех, введенных пользователем в главном модуле (***x***, ***y*** или ***z***).

14. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 3 вещественных числа: ***x***, ***y***, ***z***. Второй модуль должен содержать функцию, которая выполняет табулирование функции ***y(x)=1+x+tg(x)- cos(2x)***, где ***x*** - число, которое ввел пользователь в главном модуле; за шаг приращения взять ***y***. Количество действий табулирования равно ***z***. Процесс табулирования функции необходимо выводить на экран консоли. Третий модуль программы должен включать в себя функцию, которая выводит на консоль минимальное число из трех, введенных пользователем в главном модуле (***x***, ***y*** или ***z***).

15. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 3 вещественных числа: ***x***, ***y***, ***z***. Второй модуль должен содержать функцию, которая выполняет табулирование функции ***y(x)=1-x+sin(x)-cos(1+x)***, где ***x*** - число, которое ввел пользователь в главном модуле; за шаг приращения взять ***y***. Количество действий табулирования равно ***z***. Процесс табулирования функции необходимо выводить на экран консоли. Третий модуль программы должен включать в себя функцию, которая выводит на консоль среднее арифметическое чисел, введенных пользователем в главном модуле (***x***, ***y*** и ***z***).

16. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 1 вещественное число: ***z***. Второй модуль должен содержать функцию, которая выполняет перевод числа ***z*** в более крупные единицы измерения информации (до Тб). Представить, что ***z*** изначально содержит размер в битах. Вывести на экран перевод ***z*** во всех единицах измерения информации. Третий модуль программы должен включать в себя функцию, которая выводит на консоль значение функции $N = \frac{\sqrt[5]{z+\sqrt{zx}}}{e^x + a^5 \operatorname{arctg} x}$, где вещественные числа ***a*** и ***x*** дополнительно запрашиваются у пользователя в третьем модуле.

17. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 2 целых числа: ***a*** и ***b***. Запретить ввод числа ***a*** больше 5. Второй модуль должен содержать функцию, которая создает одномерный массив в размерностью ***b*** и заполняет его случайными целыми числами в интервале [0, ***a***]. Третий модуль программы должен включать в себя функцию, которая выводит на консоль

исходный заполненный массив из второго модуля и выполняет в нем поиск, а также выводит на экран консоли все числа, которые повторяются в массиве хотя бы один раз.

18. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 2 целых числа: ***l*** *и* ***i***. Второй модуль должен содержать функцию, которая вычисляет магнитную энергию контура с током по формуле $W = \frac{1}{2} \cdot L \cdot I^2$, где L - индуктивность, I - сила тока, которые ввел пользователь в главном модуле. Вывести на экран консоли значение ***W***. Третий модуль программы должен содержать функцию, которая производит расчет площади ***S*** и периметра ***P*** прямоугольника со сторонами ***l*** и ***i*** (введенные в главном модуле). Вывести на экран консоли значения ***S*** и ***P***.

19. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 1 целое число: ***x***. Второй модуль должен содержать функцию, которая находит значение $y = 8(x-3)^6 - 7(x-3)^3 + 27$. Значение ***x*** необходимо взять из главного модуля. Вывести на экран консоли вычисленное значение ***y***. Третий модуль программы должен содержать функцию, которая вычисляет длину и площадь окружности при заданном радиусе. В качестве значения радиуса взять значение ***x*** из главного модуля. Вывести на экран консоли значение длины и площади окружности.

20. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 2 числа: ***a*** (целое) и ***b*** (дробное, в интервале [0.1; 1.0]). Второй модуль должен содержать функцию, которая табулирует математическую функцию $4^a + 7a - 3$ с шагом ***b***. Процесс табулирования функции необходимо выводить на экран консоли. Третий модуль программы должен включать в себя функцию, которая вычисляет квадратный корень всех результатов вычислений функции второго модуля и выводит на экран консоли.

21. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 3 целых числа: ***a, b, c***. Второй модуль должен содержать функцию, которая находит площадь треугольника по формуле Герона $S_\Delta = \sqrt{p(p-a)(p-b)(p-c)}$. Значения ***a, b*** и ***c*** необходимо взять из главного модуля. Значение ***p*** (полупериметр) необходимо рассчитать. Вывести на экран значение ***S*** и ***p***. Третий модуль программы должен включать в себя функцию, которая создает одномерный массив размерностью [***a***] и заполняет его случайными числами в диапазоне [***b, c***]. Заполненный массив вывести на экран консоли.

22. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя ***имя*** (name) и ***дату рождения*** (date). Второй модуль должен содержать функцию, которая вычисляет возраст пользователя на текущую дату, исходя из его даты рождения и выводит полученный возраст на экран консоли. Третий модуль программы должен включать в себя функцию, которая выводит категорию пользователя, исходя из вычисленных данных во втором модуле:

Категория пользователя	***Возраст (лет)***
Грудной ребенок	0-1
Раннее детство	1-2
Детство	3-12
Подросток	13-16
Юноша	17-21

Средний возраст	22-60
Пожилой	61-75
Старик	76-90
Долгожитель	от 91

Категорию пользователя вывести на экран консоли.

23. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 1 целое число: ***k***. Второй модуль должен содержать функцию, которая определяет, сколько целых часов (***H***) и целых минут (***M***) прошло с начала суток, если прошло ***k*** секунд (значение ***k*** необходимо взять из главного модуля). Вывести на экран фразу: «Это ***H*** часов и ***M*** минут». Третий модуль программы должен вычислять рентабельность работы предприятия (***рент***) за месяц по формуле ***рент = прибыль/себестоимость * 100%***, если себестоимость продукции в текущем месяце уменьшилась по сравнению с прошлым на 5% . Значение прибыли (***k***) взять из главного модуля, а значение себестоимости за прошлый месяц попросить пользователя ввести с экрана. Вывести на экран консоли полученное значение ***рент***.

24. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя ввод трех значений логических переменных (тип булево): ***X, Y, Z***. Второй модуль должен содержать функцию, которая вычисляет истинность логического выражения $(\neg X \wedge Z) \vee (Y \wedge \neg Z)$. Полученное значение вывести на экран консоли. Третий модуль программы должен включать в себя функцию, которая определяет, является ли истинным выражение $X \wedge \neg(\neg Y \vee Z) \vee Y$. Вывести на экран консоли сообщение вида: "Истинно" или "Ложно".

25. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя ***имя*** (name) и ***дату рождения*** (date). Второй модуль должен содержать функцию, которая вычисляет ***возраст*** пользователя на текущую дату, исходя из его даты рождения и выводит на экран консоли вычисленный возраст. Третий модуль программы должен включать в себя функцию, которая запрашивает у пользователя ***пол*** (gender) и в зависимости от введенного значения, выводит на экран консоли приветствие в формате "*Здравствуй, дорогой Александр!*" или "*Здравствуй, дорогая Анна!*".

26. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 2 числа: ***a*** (целое) и ***b*** (дробное, в интервале [0.10; 1.00]). Второй модуль должен содержать функцию, которая табулирует математическую функцию $4^a + 4a - 5$ с шагом ***b***. Процесс табулирования функции необходимо выводить на экран. Третий модуль программы должен включать в себя функцию, которая вычисляет квадратные корни всех результатов вычислений функции второго модуля и выводит их на экран консоли.

27. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 2 целых числа: ***x, y***. Второй модуль должен содержать функцию, которая создает матрицу ***A[x, y]*** и заполняет ее случайными целыми числами в интервале [0, 100]. Вывести на экран консоли заполненную матрицу. Третий модуль программы должен включать в себя функцию, которая меняет местами наибольшие элементы в 1 и 3 строках. Вывести на экран консоли матрицу с измененными элементами.

28. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 2 целых числа: ***x, y***. Второй модуль должен содержать

функцию, которая создает матрицу ***A[x, y]*** и заполняет ее случайными целыми числами в интервале [0, 50]. Вывести на экран консоли заполненную матрицу. Третий модуль программы должен включать в себя функцию, которая считает количество элементов матрицы, превосходящих среднее арифметическое значение элементов матрицы и выводит на экран консоли количество таких элементов.

29. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 1 целое число: ***x***. Второй модуль должен содержать функцию, которая создает квадратную матрицу ***A[x, x]*** и заполняет ее нулями. Третий модуль программы должен включать в себя функцию, которая заполняет пустую квадратную матрицу ***A*** (созданную в главном модуле) по следующему правилу: числа на диагонали, идущей из правого верхнего в левый нижний угол равны 1; числа, стоящие выше этой диагонали, равны 0; числа, стоящие ниже этой диагонали, равны 2. Полученную матрицу необходимо вывести на экран консоли.

30. Разработать консольную программу на языке программирования C++, состоящую из нескольких модулей. Основной (главный) модуль программы должен запрашивать у пользователя 1 целое число: ***n***. Второй модуль должен содержать функцию, которая выполняет расчет факториала ***!n*** с помощью рекурсии. Процесс вычисления факториала необходимо выводить на экран консоли. Третий модуль программы должен включать в себя функцию, которая вычисляет и выводит на экран все значения степени числа 2^x, где ***x*** находится в промежутке [0, n].

Критерии оценивания промежуточной аттестации: экзамен

Оценка	Критерии оценивания по экзамену
Высокий уровень «5» (отлично)	оценку «отлично» заслуживает обучающийся, освоивший знания, умения, компетенции и теоретический материал без пробелов; выполнивший все задания, предусмотренные учебным планом на высоком качественном уровне; практические навыки профессионального применения освоенных знаний сформированы.
Средний уровень «4» (хорошо)	оценку «хорошо» заслуживает обучающийся, практически полностью освоивший знания, умения, компетенции и теоретический материал, учебные задания не оценены максимальным числом баллов, в основном сформировал практические навыки.
Пороговый уровень «3» (удовлетворительно)	оценку «удовлетворительно» заслуживает обучающийся, частично с пробелами освоивший знания, умения, компетенции и теоретический материал, многие учебные задания либо не выполнил, либо они оценены числом баллов близким к минимальному, некоторые практические навыки не сформированы.
Минимальный уровень «2» (неудовлетворительно)	оценку «неудовлетворительно» заслуживает обучающийся, не освоивший знания, умения, компетенции и теоретический материал, учебные задания не выполнил, практические навыки не сформированы.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

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

Дисциплина реализуется посредством проведения контактной работы с обучающимися (включая проведение текущего контроля успеваемости), самостоятельной работы обучающихся и промежуточной аттестации.

При проведении учебных занятий по дисциплине обеспечивается развитие у обучающихся навыков командной работы, межличностной коммуникации, принятия решений, лидерских качеств (включая проведение интерактивных лекций, групповых дискуссий, ролевых игр, тренингов, анализ ситуаций и имитационных моделей, преподавание дисциплины в форме курса, составленного на основе результатов научных исследований, проводимых институтом, в том числе с учетом региональных особенностей профессиональной деятельности выпускников и потребностей работодателей).

Результат обучения считается сформированным, если теоретическое содержание курса освоено полностью; при устных собеседованиях обучающийся исчерпывающе, последовательно, четко и логически стройно излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, требующих применения знаний, использует в ответе дополнительный материал; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, обучающийся способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий.

Результат обучения считается несформированным, если обучающийся при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет задания, не демонстрирует необходимых умений, качество выполненных заданий не соответствует установленным требованиям.

Методические указания для обучающихся по освоению дисциплины на занятиях лекционного типа

Лекционный курс предполагает систематизированное изложение основных вопросов тематического плана. В ходе лекционных занятий раскрываются базовые вопросы в рамках каждой темы дисциплины. Обозначаются ключевые аспекты тем, а

также делаются акценты на наиболее сложные и важные положения изучаемого материала. Материалы лекций являются опорной основой для подготовки обучающихся к практическим занятиям и выполнения заданий самостоятельной работы, а также к мероприятиям текущего контроля успеваемости и промежуточной аттестации по дисциплине.

В ходе лекционных занятий необходимо вести конспектирование учебного материала. Возможно ведение конспекта лекций в виде интеллект-карт.

Методические указания для обучающихся по освоению дисциплины на занятиях практического типа

Практические (семинарские) занятия представляют собой детализацию лекционного теоретического материала, проводятся в целях закрепления курса и охватывают все основные разделы. Основной формой проведения семинаров и практических занятий является обсуждение наиболее проблемных и сложных вопросов по отдельным темам, а также решение задач и разбор примеров и ситуаций в аудиторных условиях.

Практические (семинарские) занятия обучающихся обеспечивают:

- проверку и уточнение знаний, полученных на лекциях;
- получение умений и навыков составления докладов и сообщений, обсуждения вопросов по учебному материалу дисциплины;
- подведение итогов занятий по рейтинговой системе, согласно технологической карте дисциплины.

Методические указания по самостоятельной работе обучающихся

Самостоятельная работа обеспечивает подготовку обучающегося к аудиторным занятиям и мероприятиям текущего контроля и промежуточной аттестации по изучаемой дисциплине. Результаты этой подготовки проявляются в активности обучающегося на занятиях и в качестве выполненных практических заданий и других форм текущего контроля.

При выполнении заданий для самостоятельной работы рекомендуется проработка материалов лекций по каждой пройденной теме, а также изучение рекомендуемой литературы.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

6. Материально-техническое и учебно-методическое обеспечение по дисциплине (модулю)

Основная литература:

1. Кувшинов, Д. Р. Программирование на с : учебное пособие для вузов / Д. Р. Кувшинов. — Москва : Издательство Юрайт, 2025. — 83 с. — (Высшее образование). — ISBN 978-5-534-21174-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/559503

2. Кудрявцева, И. А. Программирование: теория типов : учебник для вузов / И. А. Кудрявцева, М. В. Швецкий. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. —

652 с. — (Высшее образование). — ISBN 978-5-534-11088-3. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/565920

Дополнительная литература:

3. Огнева, М. В. Программирование на языке C++: практический курс : учебник для вузов / М. В. Огнева, Е. В. Кудрина, А. А. Казачкова. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2025. — 342 с. — (Высшее образование). — ISBN 978-5-534-18949-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/563618

Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС) и базы данных

Доступ к ЭБС предоставляется из любой точки, в которой имеется доступ к сети «Интернет», как на территории Института, так и вне ее (удаленный доступ).

1. Образовательная платформа «ЮРАЙТ» - URL: https://urait.ru/
2. Электронно-библиотечная система «BOOK.ru» - URL: https://www.book.ru
3. Научная электронная библиотека eLIBRARY.RU - URL: https://elibrary.ru (крупнейшая российская база научных публикаций, доступ к рефератам и полным текстам статей).
4. КиберЛенинка - URL: https://cyberleninka.ru (научная электронная библиотека открытого доступа).

Информационные справочные системы

1. Справочная правовая система «Консультант Плюс» - доступ по локальной сети с компьютеров библиотеки и компьютерных классов (актуальная база законодательства РФ, в т.ч. в сфере образования и социальной защиты).
2. Справочная правовая система «Гарант» - доступ по локальной сети с компьютеров библиотеки и компьютерных классов (актуальная база законодательства РФ, включая сферы образования и социальной защиты).

Профессиональные сообщества и IT-порталы

1. Habr - URL: https://habr.com/ (крупнейшее русскоязычное сообщество IT-специалистов, статьи, новости, обсуждения).
2. Stack Overflow - URL: https://stackoverflow.com/ (международный ресурс для программистов, вопросы и ответы).
3. GitHub - URL: https://github.com/ (платформа для хостинга кода, совместной разработки и открытых проектов).
4. CodeProject - URL: https://www.codeproject.com/ (статьи, примеры кода, обсуждения для разработчиков).
5. Microsoft Learn - URL: https://learn.microsoft.com/ (бесплатные учебные материалы и документация по продуктам Microsoft, .NET, Azure и др.).
6. MDN Web Docs - URL: https://developer.mozilla.org/ (ресурс для **веб-разработчиков с документацией по HTML, CSS, JavaScript и API).**

Научные базы данных зарубежных издательств (открытый доступ)

7. IEEE Xplore - URL: https://ieeexplore.ieee.org/ (доступ к книгам, статьям и материалам конференций по компьютерным наукам, электротехнике и информационным технологиям).
8. ACM Digital Library - URL: https://dl.acm.org/ (библиотека статей и материалов конференций Ассоциации вычислительной техники).

9. SpringerLink - URL: https://link.springer.com/ (книги и журналы издательства Springer, включая серии по информатике).

10. Wiley Online Library - URL: https://onlinelibrary.wiley.com/ (коллекция журналов и книг по направлению «Computer Science & Information Technology»).

Образовательные платформы и онлайн-курсы

11. Национальная платформа «Открытое образование» - URL: https://openedu.ru/ (курсы ведущих российских вузов).

12. Stepik - URL: https://stepik.org/ (российская образовательная платформа с курсами по программированию и информатике).

13. Intuit - URL: https://intuit.ru/ (национальный открытый университет, курсы по информационным технологиям).

Официальные органы государственной власти и управления

1. Министерство науки и высшего образования Российской Федерации - URL: https://www.minobrnauki.gov.ru/.
2. Министерство просвещения Российской Федерации - URL: https://edu.gov.ru/.
3. Министерство труда и социальной защиты Российской Федерации - URL: https://mintrud.gov.ru/.
4. Федеральная служба по надзору в сфере образования и науки (Рособрнадзор) - URL: http://obrnadzor.gov.ru/.
5. Государственная система правовой информации «Законодательство России» - URL: http://pravo.gov.ru/.

Образовательные и справочные порталы

Федеральный портал «Российское образование» - URL: http://www.edu.ru/.

Российское общество «Знание» - URL: https://znanierussia.ru/.

Справочно-информационный портал «Грамота.ру» - URL: http://gramota.ru/ (русский язык и культура речи).

Образовательный портал «Учеба» - URL: http://www.ucheba.com/.

Словари и энциклопедии на Академике - URL: https://dic.academic.ru/.

Проект Государственного института русского языка им. А.С. Пушкина «Образование на русском» - URL: https://pushkininstitute.ru/.

Комплект лицензионного и свободно распространяемого программного обеспечения:

1. LibreOffice - офисный пакет
2. PDFedit – программа для работы с pdf
3. Yandex Browser – браузер
4. Менеджер архивов
5. Libre Base – программа для работы с БД
6. Inkscape – ПО для компьютерной графики
7. DIA – ПО для блока схем и диаграмм
8. 1С: Предприятие – Учебная версия
9. GiMP - Программа обработки изображений

Перечень материально-технического обеспечения включает:

учебные аудитории для проведения занятий лекционного типа, занятий практического (семинарского) типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, а также помещения для самостоятельной работы оснащенные компьютерной техникой с подключением к сети "Интернет" и доступом в электронную информационно-образовательную среду Института.

Наименование помещения.	Адрес

Перечень основного оборудования	
Учебная аудитория № 217 Учебная аудитория для проведения занятий лекционного и практического типа, выполнения курсовых работ, групповых и индивидуальных консультаций, текущего контроля, промежуточной и итоговой аттестации). Оборудование: рабочее место преподавателя (1); рабочие места обучающихся (36); ноутбук с лицензионным ПО (LibreOffice) и возможностью выхода в интернет (1); мультимедийное оборудование (1); доска учебная (1); книжный шкаф (1); сплит-система(1); учебно-наглядные пособия; доступ в электронную информационно-образовательную среду Института.	350002, Краснодарский край, г. Краснодар, Центральный внутригородской округ, ул. им. Леваневского, д. 187/1
Аудитория № 218 Помещение для самостоятельной работы обучающихся Оборудование: рабочие места обучающихся (17); персональный компьютер с лицензионным ПО и возможностью выхода в Интернет (17); книжный шкаф (1); сплит-система (1); учебно-наглядные пособия; доступ в электронную информационно-образовательную среду Института.	350002, Краснодарский край, г. Краснодар, Центральный внутригородской округ, ул. им. Леваневского, д. 187/1